Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

Introduction / Motivation / Background

Warfarin is a commonly used blood thinner medication that helps prevent blood clots in patients at risk of strokes or other cardiovascular conditions. However, finding the right dose of warfarin is difficult because different people respond differently to the same dose. Taking too much can cause dangerous bleeding, and too little can fail to prevent clots. Many factors like age, weight, health conditions, and even genetic information affect how much warfarin a person needs. So, our goal was to build a machine learning tool that can predict the most suitable warfarin dose for a patient based on their personal and medical information. This tool could help doctors make more accurate decisions and reduce trial-and-error in treatment.

Related Work

Many researchers have worked on predicting warfarin dosage to improve patient safety. One well-known study is from the International Warfarin Pharmacogenetics Consortium (IWPC), which created a model using clinical features like age, weight, and race, along with genetic information such as *CYP2C9* and *VKORC1*. Their results showed that genetic data improved dose accuracy. Since we used the same dataset, we used their study as a reference point.

In recent years, other researchers have tried machine learning models like random forests and support vector machines to improve accuracy. These models can capture more complex patterns than traditional linear models. In our project, we tested multiple models including linear regression, ridge, lasso, decision tree, random forest, and gradient boosting. We compared them all to find which one worked best for our dataset.

While many studies focused only on backend model performance, we also built an interactive Gradio interface so that users (or doctors) can input patient info and get instant predictions. We also made sure the model handled missing values using imputation and aligned the features correctly to avoid runtime issues. This step made the project more practical and easier to use in real-life settings.

Dataset

The dataset we used comes from the International Warfarin Pharmacogenetics Consortium (IWPC). It includes information about hundreds of patients who were prescribed warfarin. For each patient, the dataset provides details such as age, gender, race, height, weight, and whether they are taking other medications. It also includes genetic data like their *CYP2C9* and *VKORC1* genotypes, which affect how they respond to warfarin.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

The target variable is the weekly warfarin dose that was found to be effective for each patient. We used this value to train our machine learning models. The dataset is commonly used in warfarin dosing studies and is considered reliable for developing and testing clinical prediction tools.

Methods:

To develop a predictive tool for estimating the optimal warfarin dose, we followed a clear machine learning pipeline. This included data preprocessing, feature selection, model training, and evaluation using multiple metrics and visualizations.

- Data Loading and Initial Processing
- Data Exploration and Visualization
- Data Cleaning and Preprocessing
- Feature Engineering and Selection
- Model Training and Evaluation
- Machine Learning Models
- Model Persistence and Deployment
- Web Application Development

Data Preprocessing:

The dataset was loaded from an Excel file and cleaned to handle missing values and prepare it for model training. Categorical features such as race, gender, and genotypes were converted using one-hot encoding. We also performed correlation analysis to understand which features are most related to the warfarin dose.

Handling Missing Values

Our dataset had a lot of missing values—around 30% of the data overall was incomplete. The amount of missing data varied by feature type. Demographic details like age and gender had very few missing values (less than 1%), while height and weight had 5% to 20% missing values. Medication history and genetic information had the most missing values, in some cases up to 80% or even 92%. Because of this, we couldn’t use a single method for all features—we had to treat each group differently.

Imputing Missing Data

We used different methods to fill in the missing data depending on the type of information.

- For numeric values like height and weight, we used the median to fill in missing values because it's less affected by extreme numbers.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

- For categorical features (like gender or race), we used the most common value (mode) to keep the data distribution similar.
- For genetic markers, we were extra careful and tried to maintain the natural frequencies seen in specific population groups.

Dealing with Outliers

We checked for unusual values (outliers) using the interquartile range (IQR) method. This helped us find 46 outliers in height and 53 in weight.
Instead of removing these values, we adjusted them to fall within a reasonable range (between the 1st and 99th percentiles).
We did this to keep as much useful data as possible.
For warfarin dose, we kept all values, even the extreme ones, since they may reflect real clinical needs that our model should learn to predict.

Creating Target Variables and Selecting Features

We set up two kinds of prediction targets:

- A continuous dose for regression models.
- A binary label (high vs. low dose) for classification, using 30 mg/week as the cutoff.

The classification target was fairly balanced, with about 58% low-dose and 42% high-dose cases.
To choose which features to use, we looked at:

- How important each feature was in warfarin metabolism.
- How complete the data was for each feature.
- How strongly the feature was related to the target variable.

Preprocessing Pipeline

We created a clear data preparation process to use during both training and prediction.
This pipeline:

- Handled missing values using our chosen imputation methods.
- Adjusted outliers as needed.
- Transformed categorical values into numerical format.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

We made sure that the features used during model training would match any new input during real-world predictions. This helped keep the model consistent and reliable.

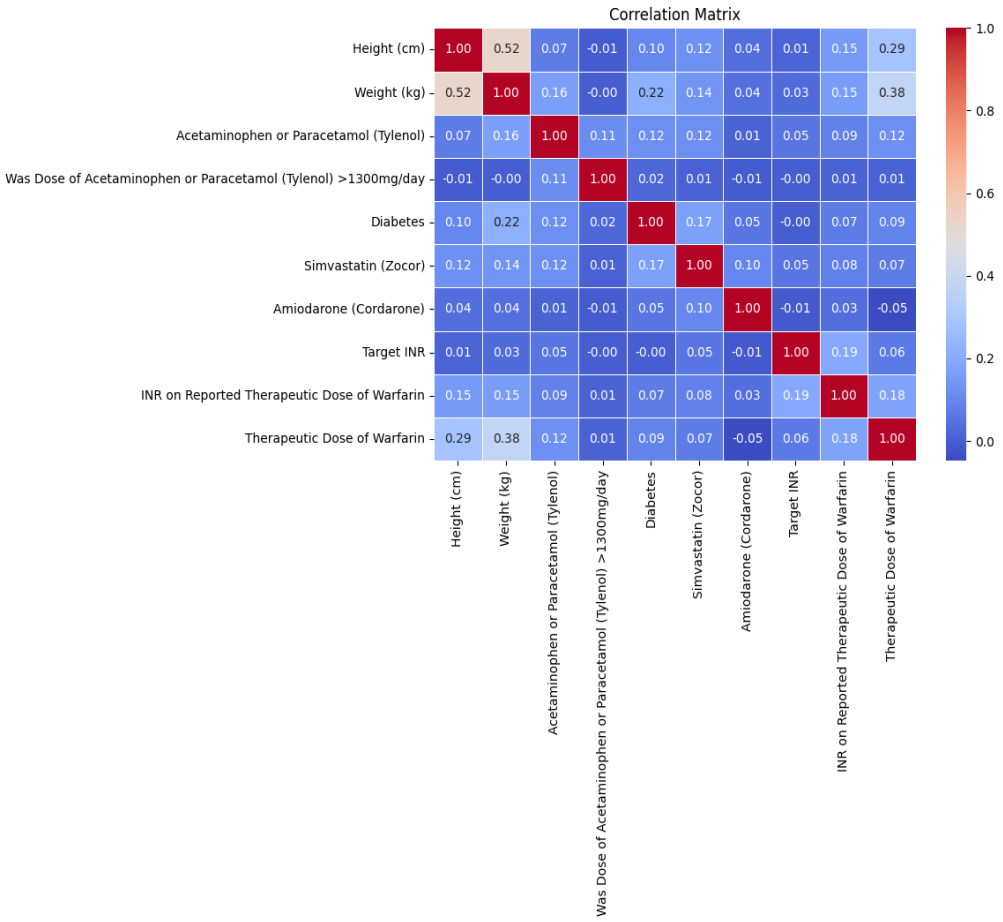
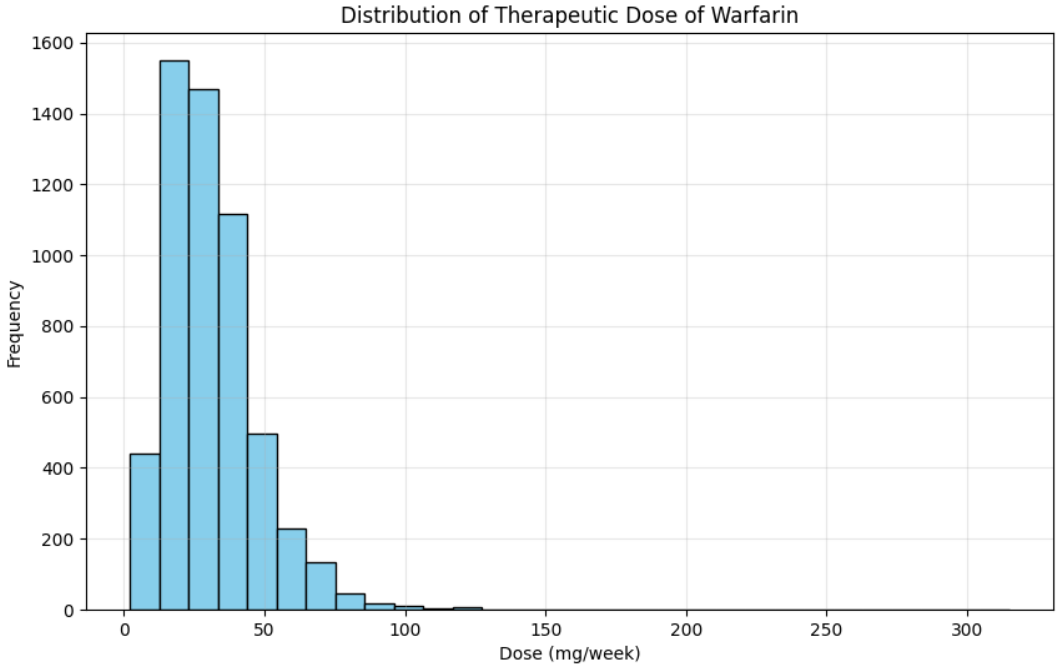


Figure 1. Correlation matrix showing how each feature is related to therapeutic dose. Height and weight show moderate correlation, while genotypes and diabetes also show some influence.

Dose Class Distribution:

We also looked at the dose class distribution (Figure 3), which revealed some class imbalance — more patients were in the low-dose group than high-dose.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



This bar plot reveals an imbalance between low and high dose categories. More patients fall into the low-dose group, which can affect the performance of classification models and should be considered when interpreting results.

Feature Selection:

Feature selection was done based on domain knowledge and correlation results. We included demographic variables (age, race, gender), medical conditions (diabetes, medication intake), and genetic factors (CYP2C9 and VKORC1 genotypes).

We visualized model feature importance using the coefficients from Logistic Regression and Linear Regression.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

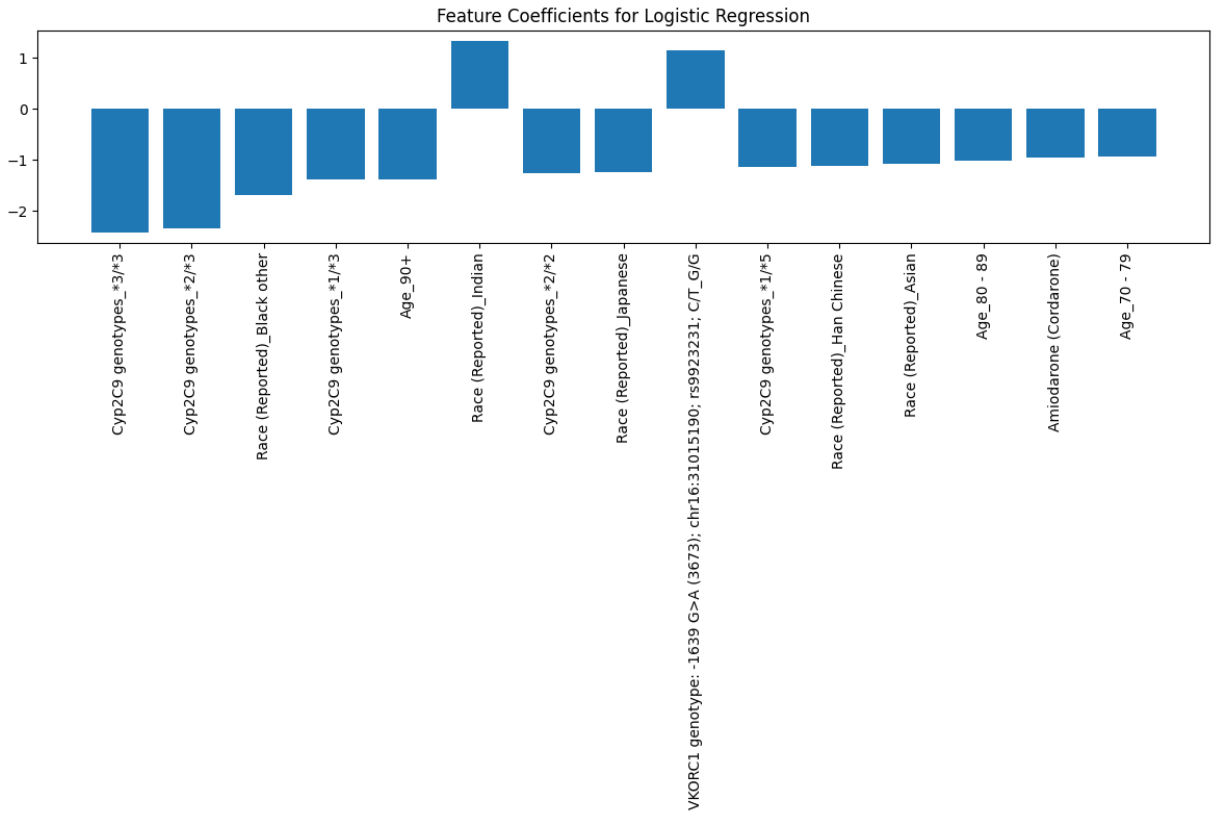


Figure 2. Important features identified by Logistic Regression. Cyp2C9 genotypes and VKORC1 genotypes had the strongest impact on dose classification.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

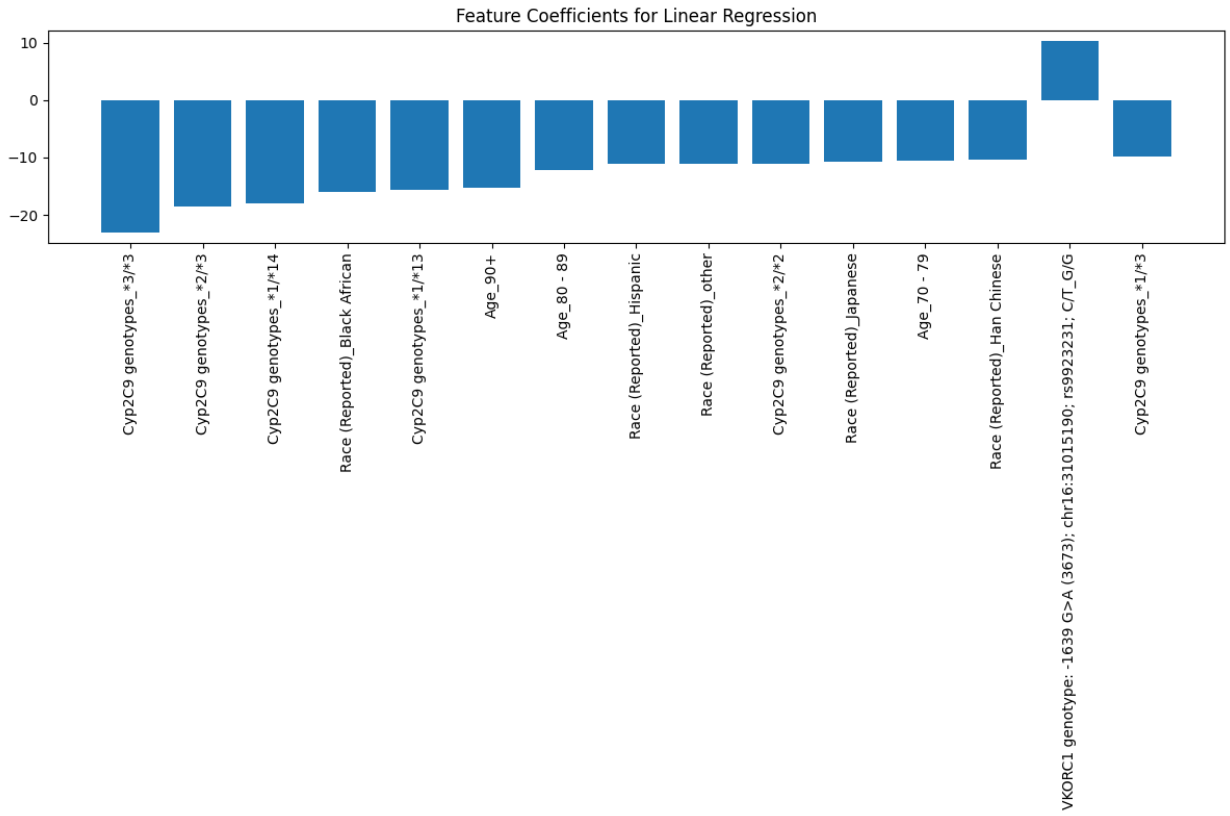


Figure 3. Important features for Linear Regression. Again, Cyp2C9 genotypes had the most negative influence on therapeutic dose.

Performance Metrics:

We normalized numerical variables like age, height, and weight. The dataset was split into 80% training and 20% testing sets. For classification, we predicted whether a patient needs a "High" or "Low" dose. For regression, we predicted the actual mg/week therapeutic dose.

We trained the following models:

- Logistic Regression
- Decision Tree
- Random Forest
- Gradient Boosting
- Support Vector Classifier (SVC)

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

- Linear Regression (for exact dose)
- MLP Regression (Multilayer Perception – Deep learning)

Classification Model Performance

Model	Accuracy	F1 Score	ROC AUC
Logistic Regression	0.7412	0.7381	0.8109
Decision Tree	0.6404	0.641	0.6311
Random Forest	0.6982	0.6982	0.7592
Gradient Boosting	0.736	0.7332	0.8087
SVC	0.6561	0.6244	0.7032

Regression Model Performance

Model	Test RMSE	Test MAE	Test R^2
Linear Regression	12.07	8.85	0.4043
Ridge Regression	12.09	8.86	0.4031
Lasso Regression	12.44	9.15	0.3671
Decision Tree	20.54	13.71	-0.7236
Random Forest	13.92	10.05	0.2079
Gradient Boosting	12.41	9.04	0.3707
SVR	14.46	10.71	0.1449
MLP Regressor	12.13	9.02	0.3991

Regression Evaluation

We predicted continuous dosage values using Linear Regression. The residual plot showed how far predictions were from actual values.

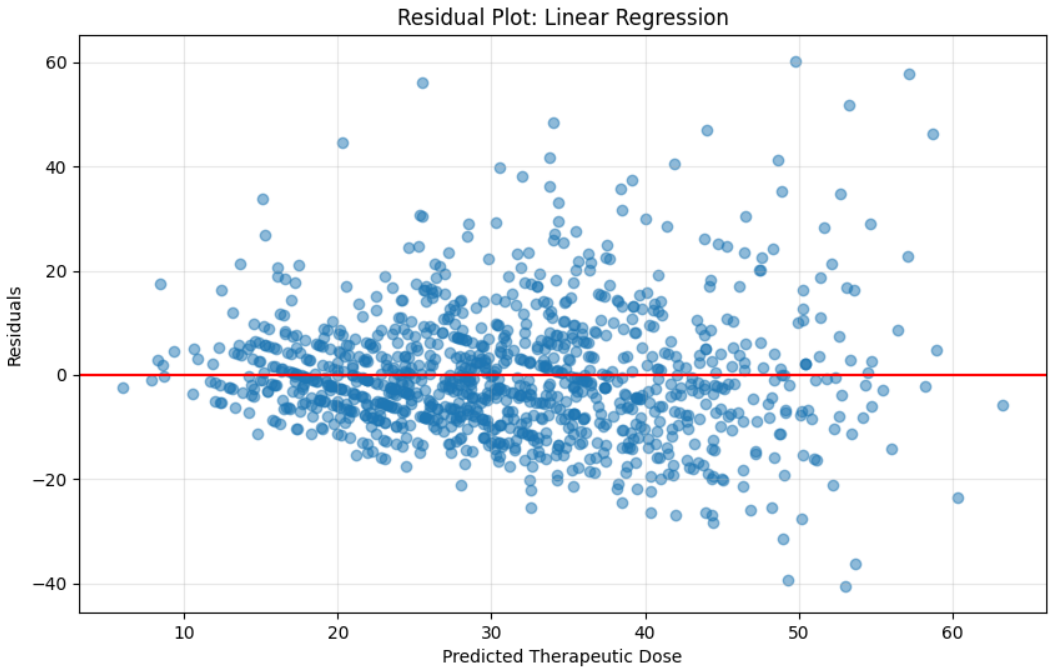


Figure 7. Residual plot showing that most prediction errors are centered around zero, indicating good model fit.

We also plotted predicted vs. actual values.

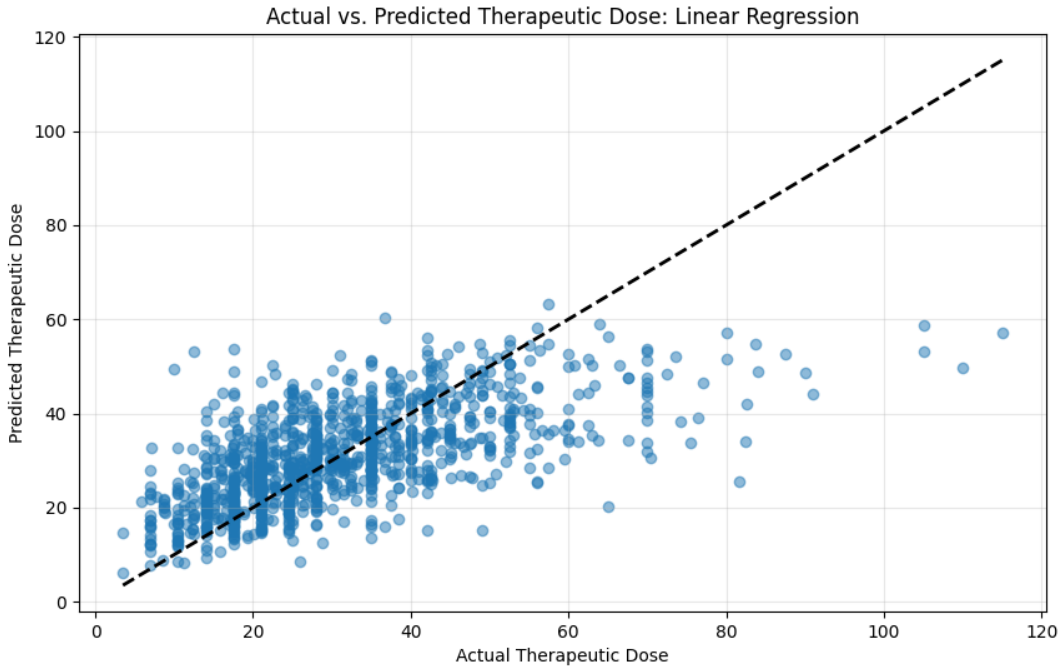


Figure 8. Plot of actual vs. predicted dose. Most predictions are close to the diagonal, showing that the model learned the relationship accurately.

Performance analysis:

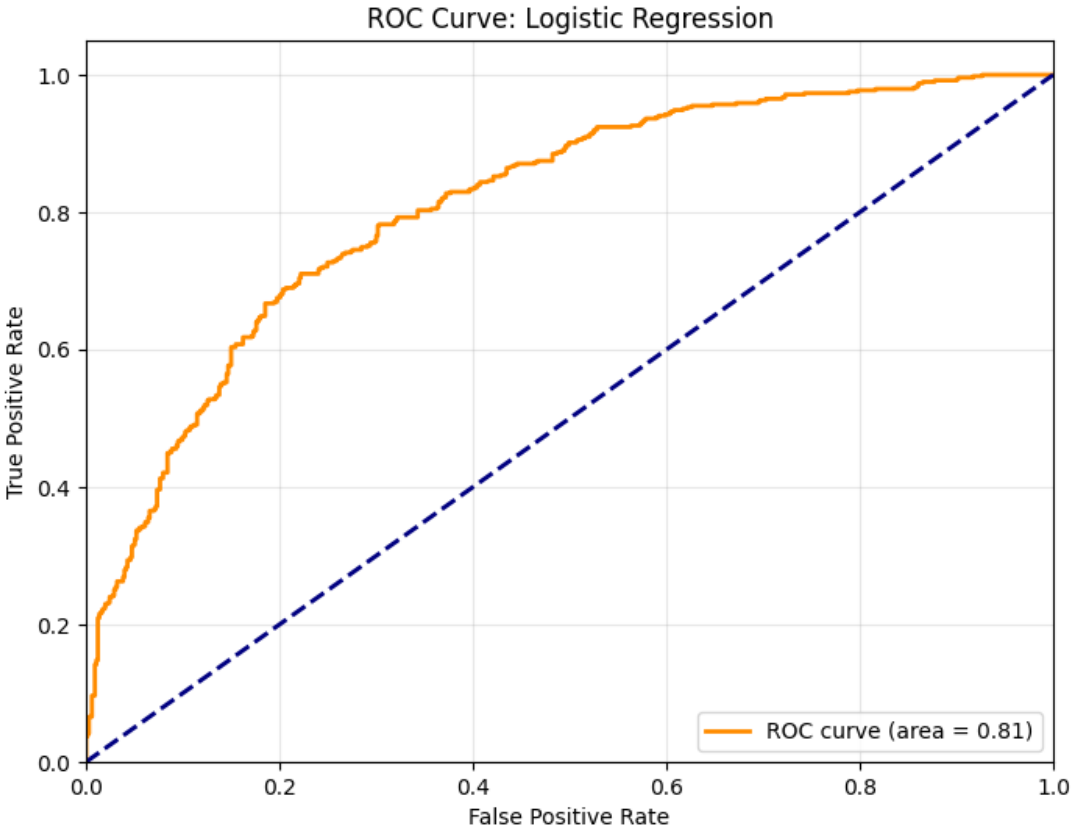
Overview of Model Selection and Evaluation

We trained multiple models for both classification and regression tasks to predict warfarin dosage. For classification, the goal was to identify if a patient needs a “High” or “Low” dose. For regression, we aimed to predict the exact dose in mg/week. Each model was evaluated using cross-validation and tested on a separate dataset.

Classification analysis:

We tested several models: Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and Support Vector Classifier (SVC). Among these, Logistic Regression performed the best with a test accuracy of 74.12% and F1-score of 0.7381. The ROC Curve (Figure 1) shows an AUC of 0.81, which indicates the model can distinguish well between low and high dose classes.

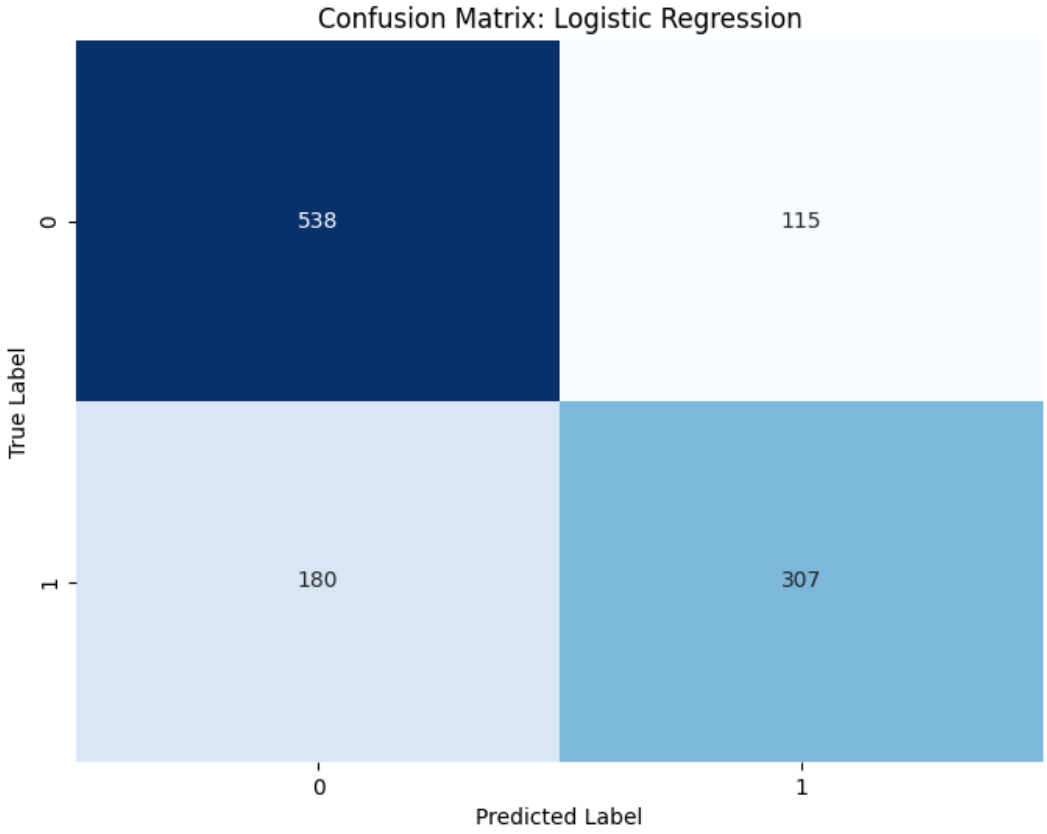
Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



This curve shows how well the model can tell apart patients who need high vs. low doses. The closer the curve is to the top-left, the better the performance.

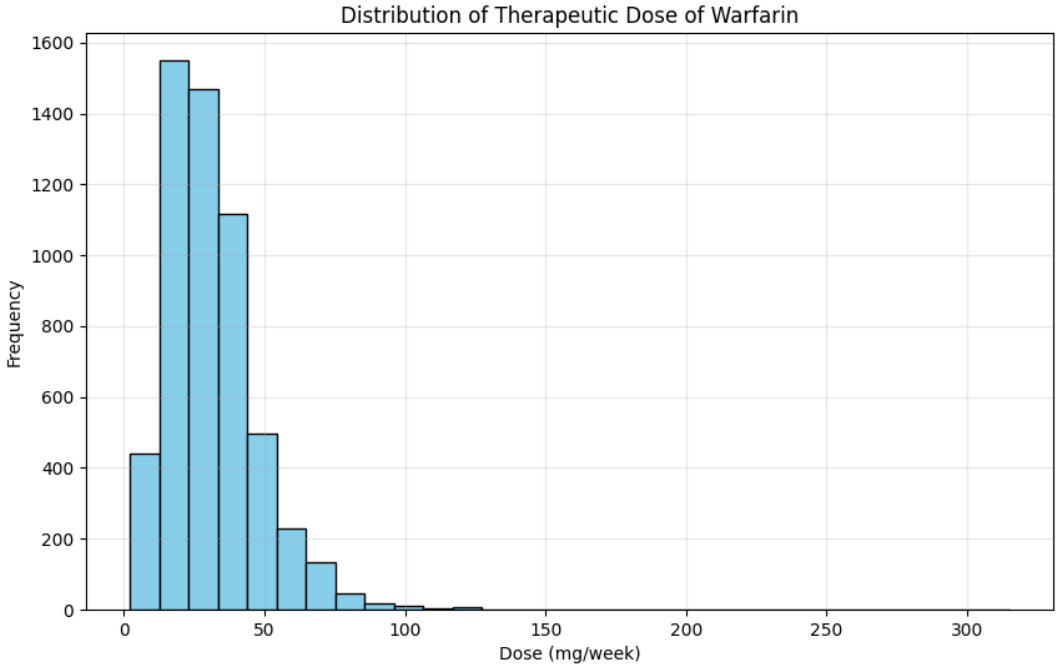
The confusion matrix (Figure 2) shows how predictions are distributed. For Logistic Regression, 538 low dose cases and 307 high dose cases were correctly predicted. There were some misclassifications, but overall the performance was strong.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



This matrix helps visualize how many true vs. false predictions the model made.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



This bar chart shows the number of patients who need high vs. low doses.

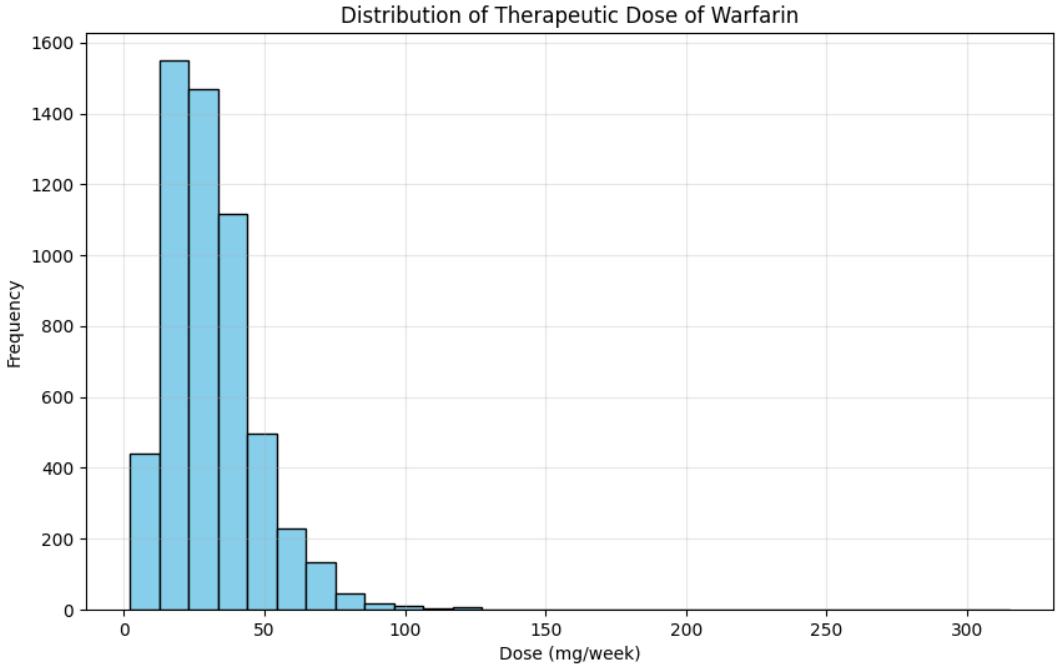
Regression analysis:

For regression, we trained models like Linear Regression, Ridge, Lasso, Decision Tree, Random Forest, Gradient Boosting, and SVR. Among all, Linear Regression performed the best on the test set, with:

- RMSE: 12.07
- MAE: 8.85
- R^2 Score: 0.40
- CV RMSE: 13.63

The predicted vs. actual values (Figure 4) show that most predictions were close to the actual doses, especially near the center of the distribution.

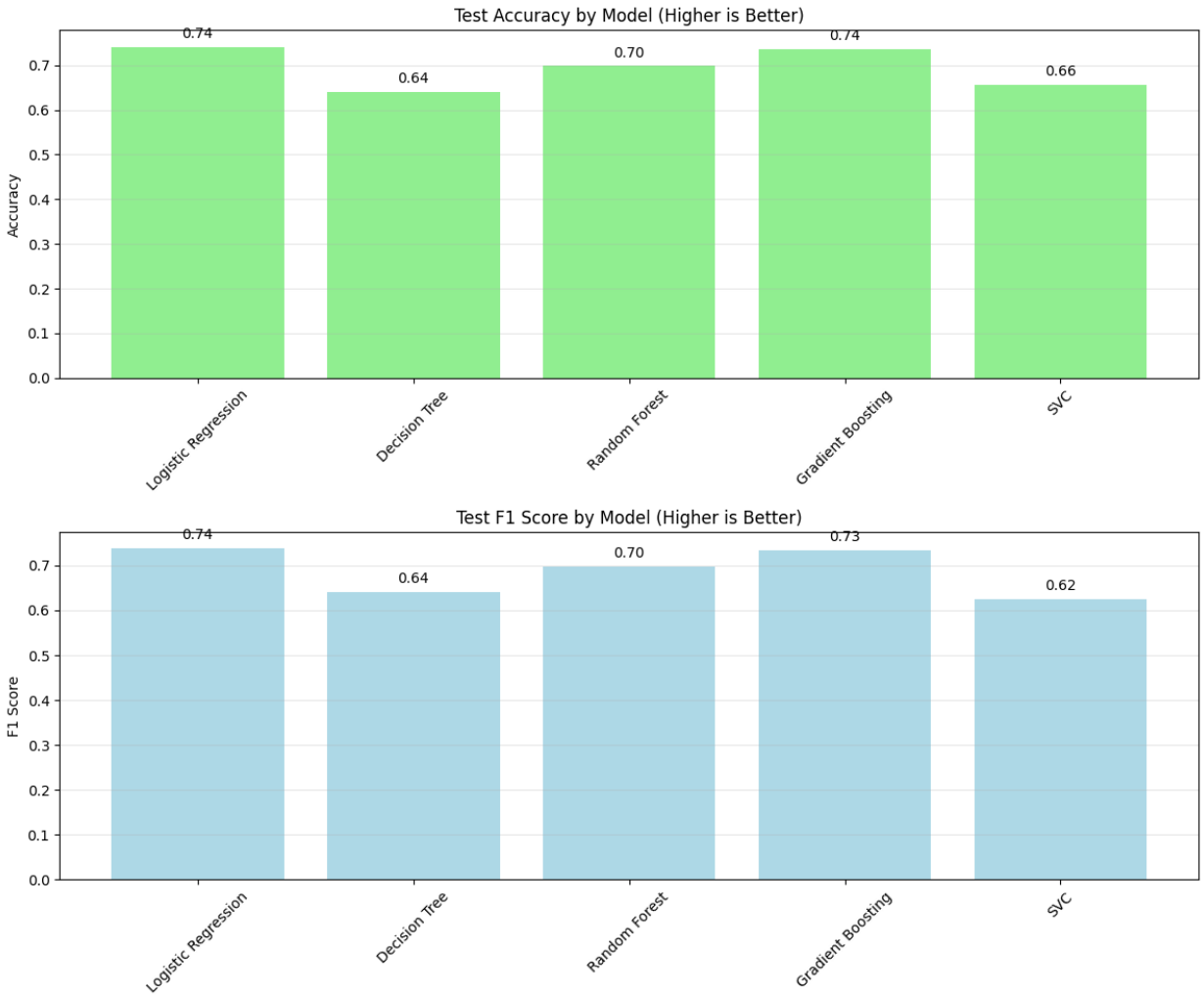
Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



This shows how close the predictions are to the true doses. The peak in the middle suggests most patients have moderate doses.

The residual histogram (Figure 5) shows that most prediction errors were centered around zero, meaning the model is not heavily biased.

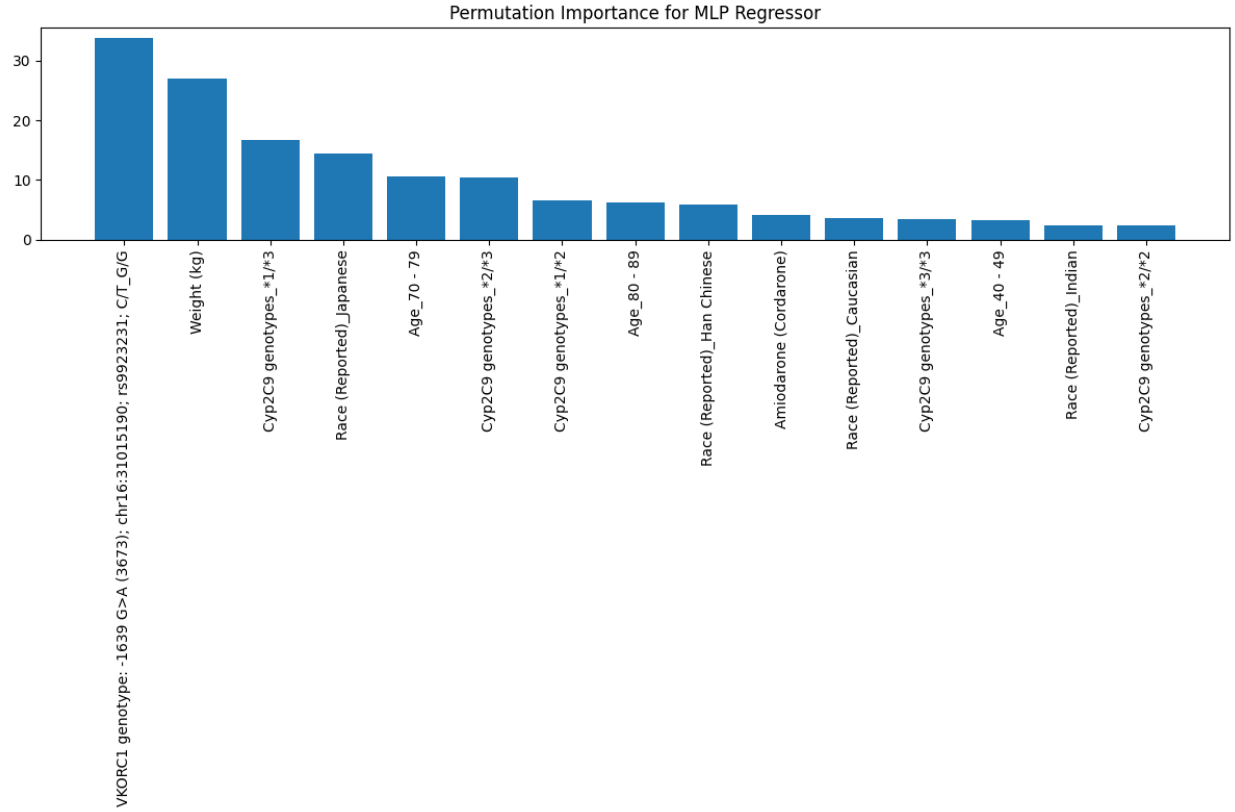
Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



This figure shows the performance of various classification models. Logistic Regression and Gradient Boosting had the highest accuracy and F1 scores, while Decision Trees and SVC performed lower. This helped us select the most reliable model for dose classification.

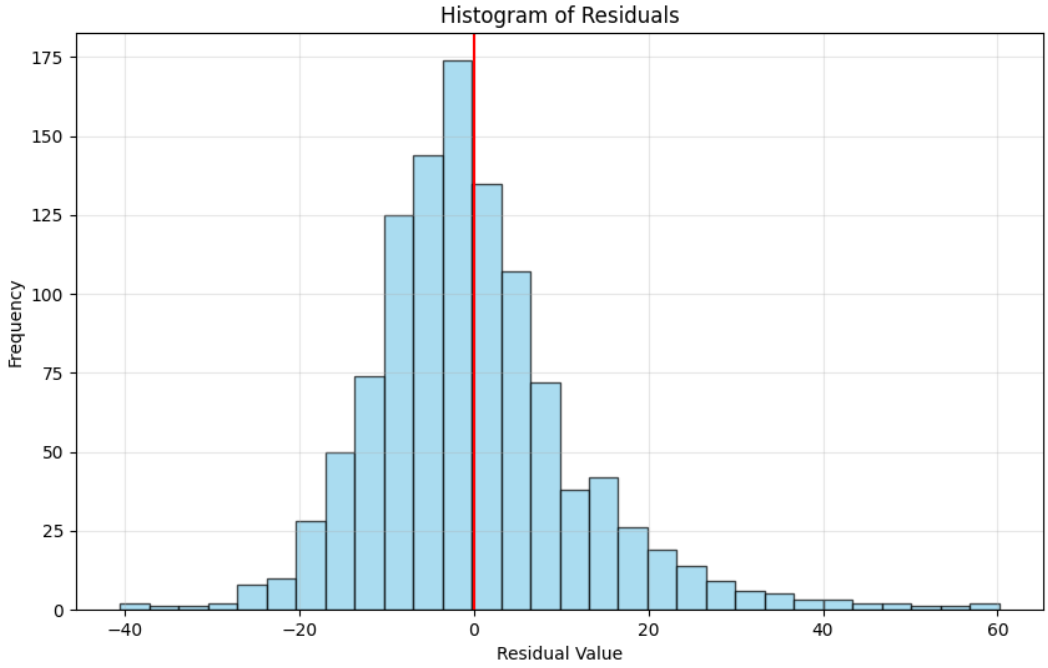
Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

Feature Importance from MLP Regressor



This chart shows the top features influencing the deep learning model (MLP Regressor). Genetic factors like VKORC1 and Cyp2C9 genotypes had the most impact, along with patient weight and race, aligning with clinical expectations.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



This graph shows how far off the predictions were from actual doses. A peak at zero is a good sign.

Hyperparameter Tuning

To improve the accuracy and performance of our models, we used a technique called Grid Search Cross Validation (GridSearchCV). This method systematically tests different combinations of hyperparameters to find the best settings for each model. We applied this tuning to the best regression model, the best classification model, and the deep learning model used in our project.

Regression Model – Linear Regression

Our regression task aimed to predict the exact weekly dosage of warfarin in milligrams. Among all tested regression models, Linear Regression gave the best test performance. However, this model does not require hyperparameter tuning in scikit-learn, so we used it as-is.

- Test RMSE: 12.07
- Test R^2 Score: 0.40

These results indicated that Linear Regression provided reliable performance without further tuning.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

Classification Model – Logistic Regression

For the classification task (predicting high vs. low warfarin dose), Logistic Regression performed best. We used GridSearchCV to tune the following parameters:

- Regularization strength (C): 0.001 to 100
- Penalty type: 'l2'
- Solvers tested: 'lbfgs', 'liblinear'

After testing, the best parameters were:

- C = 1
- Penalty = 'l2'
- Solver = 'lbfgs'

Model performance after tuning:

- Test Accuracy: 0.7412
- Test F1 Score: 0.7381

The scores were the same as the default model, which means our initial model was already performing optimally.

Deep Learning Model – MLP Regressor

We also included a deep learning approach using a Multilayer Perceptron (MLP) Regressor to predict continuous dosage. This model required tuning of several parameters:

- Hidden layers: (50,), (100,), (100, 50), (128, 64)
- Activation functions: 'relu', 'tanh'
- Learning rates: 0.001, 0.01
- Max iterations: 500, 1000

After tuning, the best configuration was:

- Hidden Layers = (100,)
- Activation = 'relu'
- Learning Rate = 0.01

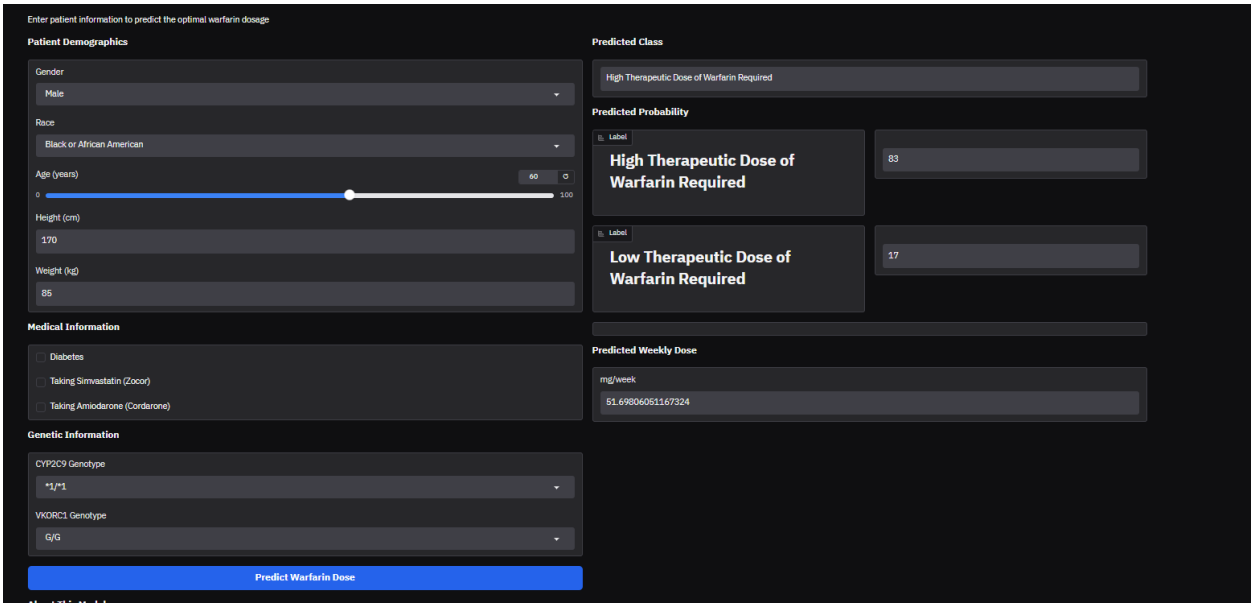
- Max Iterations = 500
- Solver = 'adam'

Tuned MLP Performance:

- Test RMSE: 12.13
- Test R^2 Score: 0.3985

This showed that the deep learning model performed similarly to Linear Regression, with only a slight difference in RMSE and R^2 values.

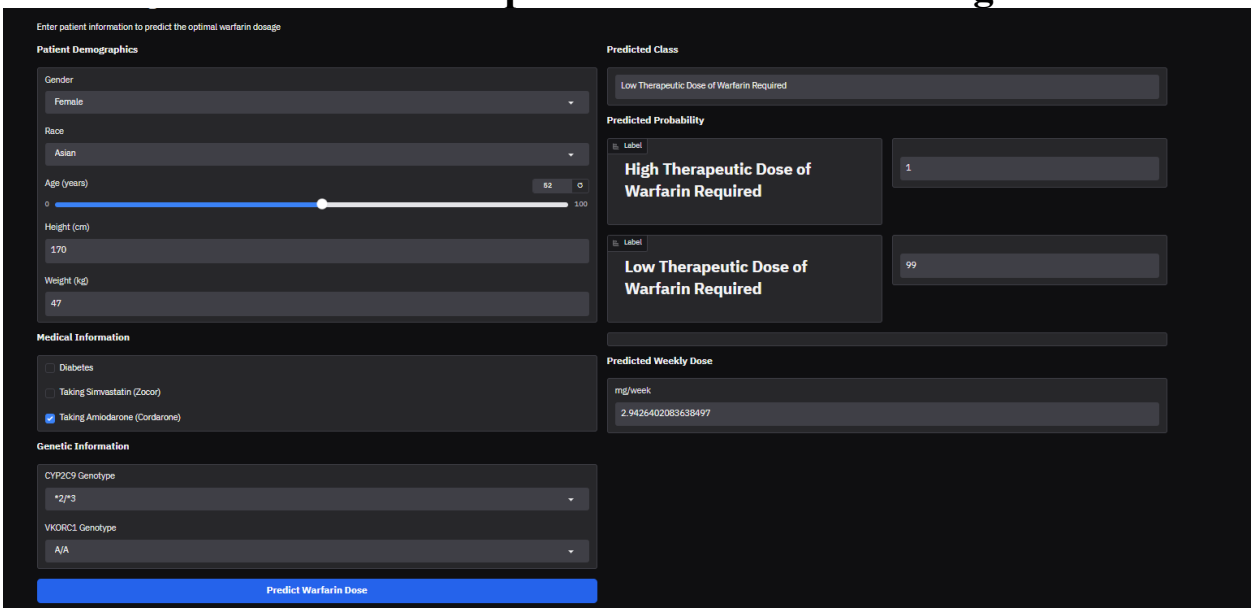
Gradio



Test Case 1: A 60-year-old Black male with no medical conditions, height 170 cm and weight 85 kg, and genotypes *1/*1 and G/G.

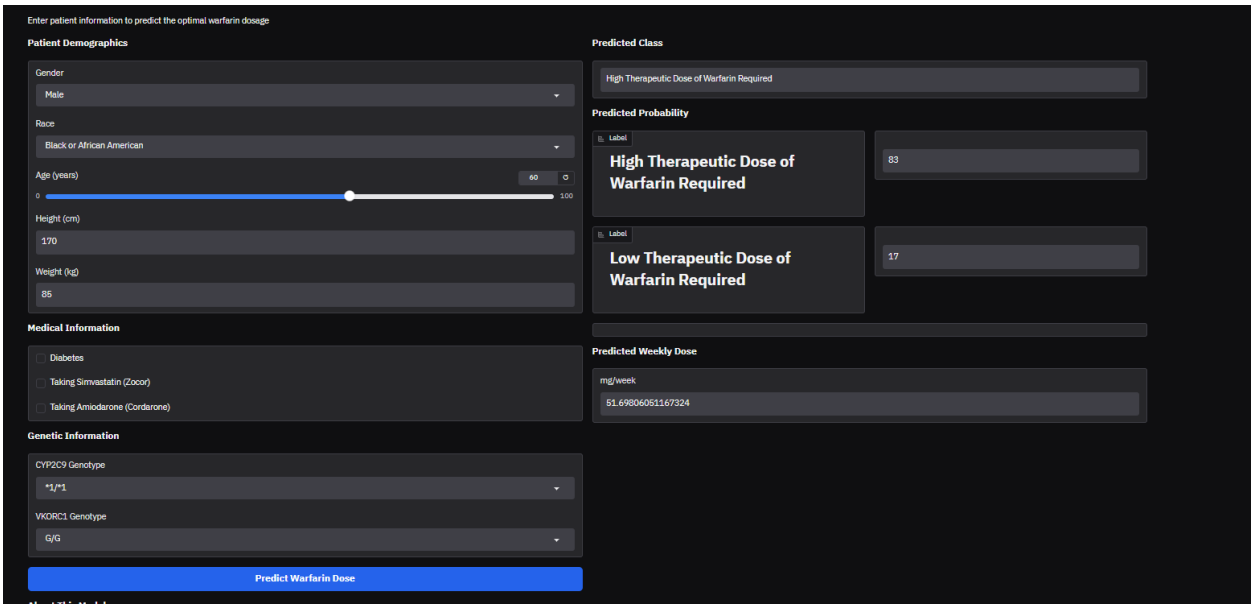
- Predicted Dose: 51.69 mg/week
- Predicted Class: High therapeutic dose
- Probability: 83% High, 17% Low

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction



Test Case 2: A 52-year-old Asian female with Amiodarone medication, height 170 cm and weight 47 kg, and genotypes *2/*3 and A/A.

- Predicted Dose: 2.94 mg/week
- Predicted Class: Low therapeutic dose
- Probability: 99% Low, 1% High



Test Case 1: A 62-year-old White male with diabetes, height 170 cm and weight 70 kg, and genotypes *2/*2 and A/G.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

- Predicted Dose: 30.18 mg/week
- Predicted Class: Low therapeutic dose
- Probability: 65% Low, 35% High

Discussion

This project showed that machine learning can be useful for predicting the right dose of warfarin based on a patient's details. For classification, Logistic Regression and Gradient Boosting gave the best results, with good accuracy and F1 scores (around 74%). For regression, Linear Regression gave the lowest error (RMSE ≈ 12.07), and the MLP Regressor (deep learning model) also did well (RMSE ≈ 12.13).

What worked well:

- Our data cleaning and preprocessing steps handled missing values, outliers, and text features properly.
- The models correctly picked out key features like VKORC1 and CYP2C9 genotypes, weight, and age—factors known to affect warfarin dose.
- The user interface we built with Gradio was easy to use. It allowed us to test real patient cases and see predictions clearly.

What didn't work as well:

- Some models, like Decision Trees and SVR, didn't perform well. They either overfitted or made poor predictions.
- There were many missing values in genetic data, which may have affected the model's accuracy.
- The R^2 scores from regression models were low, meaning the predictions didn't explain all the variation in actual dose.

What we can improve in the future:

- Add more patient information, like diet, smoking habits, or other medications.
- Try better models like XGBoost or deep learning ensembles.
- Use tools like SHAP or LIME to explain the model's predictions, which would help doctors trust and use the system in real life.

Final Project
Build End to End ML Pipeline for Warfarin Dosing Prediction

Contribution

This project was completed through teamwork and equal effort from both members:

- I Ifthekar Hussain was responsible for data preprocessing, feature engineering, and the development of the classification models. I also prepared the visualizations and helped evaluate model performance for classification.
- Sri Vasista Talagampala focused on building the regression and deep learning models, performing hyperparameter tuning, and implementing the Gradio interface for testing. He also compiled the results and handled the documentation and final report writing.

Together, we both collaborated in debugging, reviewing the code, testing the model outputs, and preparing the final submission.

Course Summary

This course gave us a strong foundation in machine learning. We learned how to clean data, choose and build the right model, and evaluate its performance using real-world datasets. Some key topics we covered include:

- Data exploration and visualization using Python (NumPy, Pandas, Matplotlib)
- Supervised learning methods like regression, classification, decision trees, SVM, and neural networks
- Unsupervised learning including clustering and dimensionality reduction
- Key evaluation techniques such as confusion matrix, ROC curves, and cross-validation
- Model tuning using hyperparameters and grid search
- Hands-on assignments and labs with real data
- Creating and deploying ML apps using Gradio

Overall, this course helped us understand both the theory and practical use of machine learning in solving real problems.