## **Project Report - Warfarin Dose Prediction**

**Table of Contents**

[**Project Report - Warfarin Dose Prediction** 1](#_Toc134887987)

[**Chapter 01 - Introduction** 2](#_Toc134887988)

[**Abstract** 2](#_Toc134887989)

[**Introduction** 2](#_Toc134887991)

[**Importance** 2](#_Toc134887992)

[**Current research** 3](#_Toc134887993)

[**Chapter 02 - Methods** 4](#_Toc134887994)

[**Overview of your experimental pipeline** 4](#_Toc134887995)

[**Describe how the data is obtained and processed.** 5](#_Toc134887996)

[**Describe how the data is visualized.** 5](#_Toc134887997)

[**How is the missing data handled (i.e., imputation techniques)?** 7](#_Toc134887998)

[**Describe how multicollinearity is assessed.** 8](#_Toc134887999)

[**Clearly describe input features and outputs for ML methods.** 8](#_Toc134888000)

[**Describe the machine learning methods you choose to study the topic, including a description of algorithms.** 8](#_Toc134888001)

[**Describe how the ML model/results will be evaluated.** 10](#_Toc134888002)

[**Chapter 03 - Results** 12](#_Toc134888003)

[**Model selection and Hyper parameter tuning** 12](#_Toc134888004)

[**Evaluation of Regression Models** 12](#_Toc134888005)

[**Evaluation of Classification Models** 13](#_Toc134888006)

[**How to build web applications for your ML models?** 16](#_Toc134888007)

## **Chapter 01 - Introduction**

### **Abstract**

### Due to its limited therapeutic range and potential negative consequences, it is vitally important to accurately predict the dosage of warfarin for individual patients. Clinical algorithms, pharmacogenetic testing, and machine learning techniques have all been used to enhance the accuracy of dose prediction. Several machine learning algorithms, including Stacked Generalized Regression Model, Logistic Regression, Support Vector Machines, Ensemble Learning, and Decision Trees, were utilized by the project to predict warfarin doses. Based on a patient's genetic profile and other clinical characteristics, these methods are used to generate models capable of predicting the optimal warfarin dosage for each patient. The ultimate objective is to enhance the results of treatment and safeguard patients through offering clinicians accurate and reliable dosing information.

### **Introduction**

Warfarin's limited therapeutic window and interpatient variability make it difficult to determine the right dose. Numerous research has investigated various methods for forecasting warfarin doses throughout the years, including clinical algorithms, pharmacogenetic testing, and machine learning methods.

Data mining techniques are used to explore massive datasets to create models that can precisely forecast the ideal quantity of warfarin to be delivered for every patient. Data mining, a method for drawing useful conclusions and patterns from massive amounts of data, has become more and more popular in the medical field to enhance decision-making and enhance therapeutic strategies.

### **Importance**

More than 30 million prescriptions for the medicine were written in the United States in 2004 for warfarin, the most used oral anticoagulant agent globally. Predicting the optimal warfarin dose is essential for treating individuals who need anticoagulant medication. The appropriate dosage of warfarin varies depending on the patient and is indicated for a variety of cardiovascular diseases. To maximize effectiveness while lowering dangers, healthcare professionals can personalize the treatment to each patient's needs with accurate dose prediction. The proper dosage of warfarin is required to balance the hazards of clotting and bleeding within its narrow therapeutic range.

The goal of current warfarin dose prediction research is to develop dosing regimens that are more precise and accurate. To improve the accuracy of dose prediction, studies have concentrated on a variety of aspects, including clinical algorithms, pharmacologic testing, and machine learning techniques. Predicting the correct dose prediction helps avoid under- or over-dosing, ensuring that patients get the best anticoagulant effect while minimizing side effects. Genetic differences may affect how patients react to warfarin. By integrating genetic testing and dose prediction algorithms, healthcare providers can identify patients who may require adjusted doses based on their genetic profile, leading to more personalized and effective treatment.

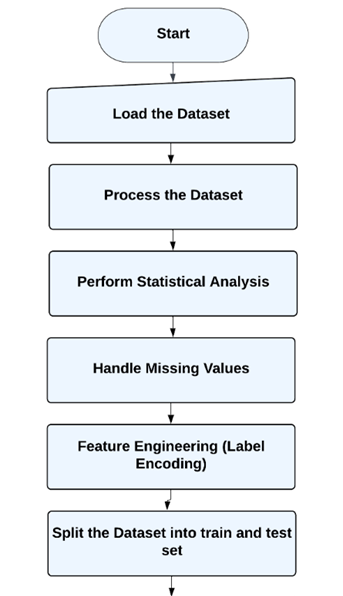
### **Current research**

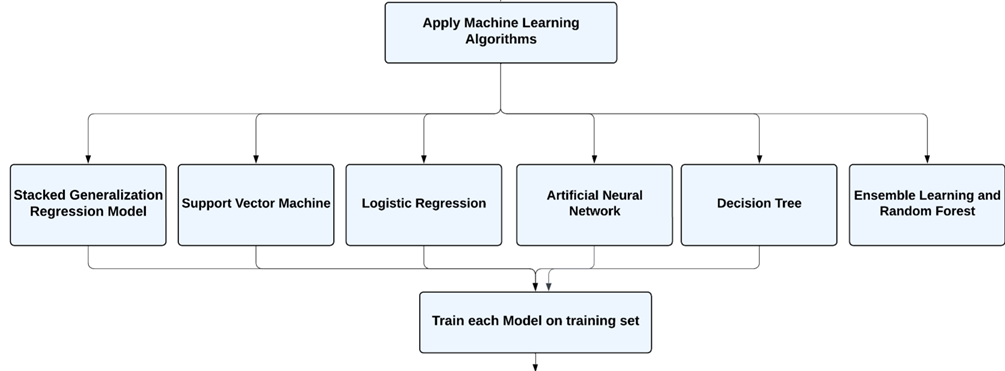
Clinical algorithms, such as the Coumadin dosing algorithm, offer initial dose recommendations based on patient characteristics and have been verified. Individualized dose is now possible because of pharmacologic testing that has shown genetic variants in the CYP2C9 and VKORC1 genes that affect warfarin sensitivity. Multiple patient characteristics are being included into machine learning techniques, such as support vector machines and neural networks, to create more sophisticated prediction models. The goal of the study is to improve treatment outcomes and patient safety by giving clinicians more accurate and trustworthy information to help them choose the best warfarin dose.

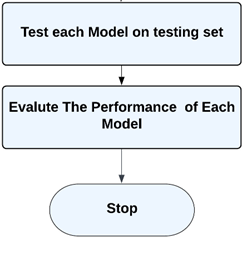
## **Chapter 02 - Methods**

### **Overview of your experimental pipeline**

To analyze and forecast the Warfarin therapeutic dose, the experimental pipeline includes data preprocessing, exploratory data analysis, feature selection, model training, and evaluation.







### **Describe how the data is obtained and processed.**

The pd.read\_excel() function of the Pandas library is used to retrieve data from an Excel file. It proceeds to be read into the Pandas DataFrame. Using the head function, the dataset is restricted to the first 5700 rows because the rest of the rows are empty.

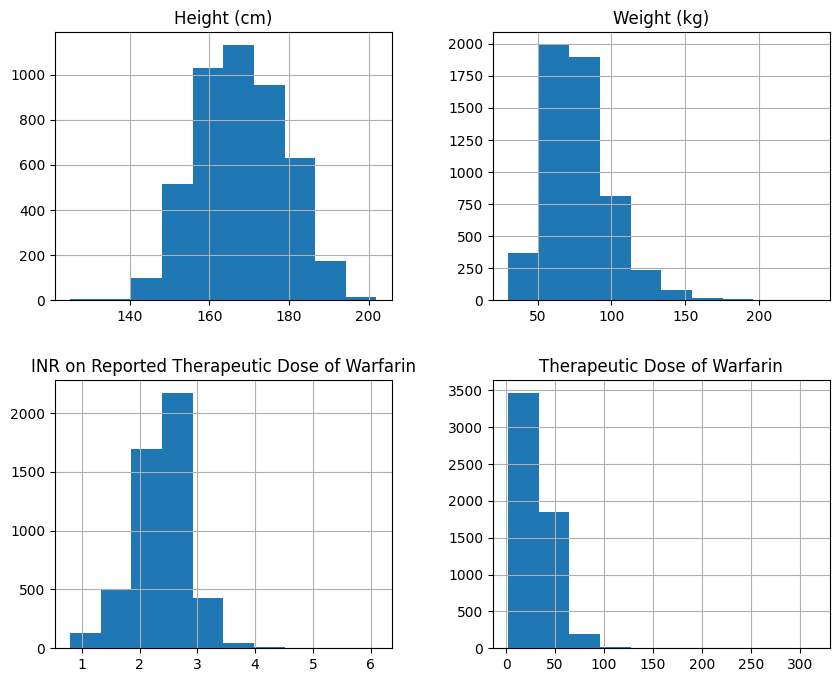
### **Describe how the data is visualized.**

Statistical analyses incorporated into our project include:

1. **Describe Method:** The describe() method is used in this code to create summary statistics, which are then printed together with the DataFrame that contains them. T computes several summary statistics that aid in comprehending the dataset's central tendency, dispersion, and shape. The function offers the following summary statistics, such as:

* Count: The number of non-null values
* Mean: the dataset average value
* Standard deviation: the measure of the spread of the dataset
* Minimum and maximum values: dataset containing smallest and largest values.
* Quantiles: The values (e.g., quartiles, percentiles) that divide the dataset into the specified parts

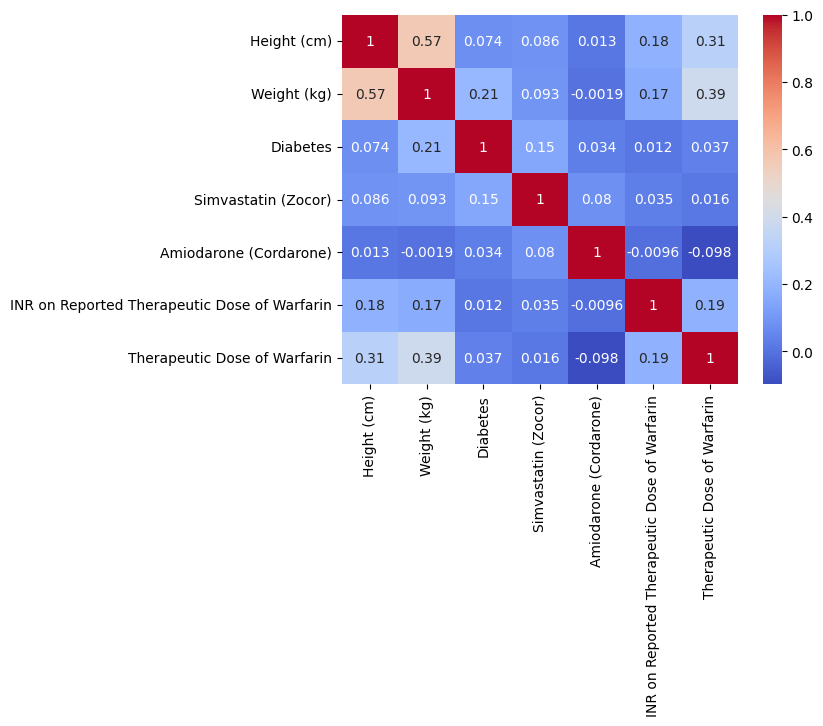
1. **Histogram Visualization:** The alternative method creates a histogram for each column of numbers in a Pandas DataFrame. A histogram is a graph that shows how a dataset is distributed. It displays the frequency of values that fall into particular data bins or ranges. The Pandas DataFrame's hist() function plots a histogram of the chosen columns.



**Figure 1 shows the histogram plots**

1. **Correlation Analysis:** The Pandas package in Python provides the statistical function df.corr() that determines the pairwise correlation between each column in a DataFrame. By default, it calculates the Pearson correlation coefficient for each pair of columns in the DataFrame. The linear link between two variables is measured by the Pearson correlation coefficient as shown in figure 2. It spans from -1 to +1, with closer values denoting a strong positive correlation and closer values denoting a strong negative correlation.

No correlation between the two variables is shown by a value of 0. A DataFrame holding the correlation coefficients for each pair of columns is the result of the df.corr() method. Since each column is fully correlated with its own, the resulting DataFrame is a symmetric matrix with 1 as the diagonal element.



**Figure 2 shows the correlation between features.**

### **How is the missing data handled (i.e., imputation techniques)? Do any of the variables have multicollinearity issues?**

Dealing with missing values is a crucial data preprocessing phase prior to constructing machine learning models. Using imputation, missing values in the specified columns are then handled. The mode is used to replace absent values in columns that are categorical with missing values. Depending on the skewness of the distribution, the median or mean imputation strategy is used to fill in missing values in numerical columns.

So, we drop only those columns whose maximum values are missing in the dataset. Filling the values that are missing with the mean or mode may create bias and skew the normal distribution of the data, especially when there are many missing values or when the missing values are not absent at random. Additionally, if the data includes outliers or is unbalanced, the mean may not accurately represent the data's central tendency. Noting that imputation and approximation are not perfect options and can introduce their own errors or biases is essential.

Before employing any imputation or interpolation method to fill in missing values in a dataset, it is crucial to thoroughly assess the results while considering the potential limitations of the method. First, we determine the correlation between the columns. By removing columns that have a low correlation with the desired variable, we can potentially reduce the level of dimensionality of the dataset and enhance the efficacy of machine learning models that we may train on the data.

### **Describe how multicollinearity is assessed.**

**Multicollinearity Assessment:** The heatmap function generates a color-coded matrix that represents the correlation coefficients between all pairs of data features. As input, the function receives the correlation matrix and returns a diagram in which the hue of each cell depicts the degree of the correlation among the respective group of variables.

For determining the correlation matrix, we can use the corr() method from the pandas library, which computes the mutual correlation between columns in a DataFrame. The resultant correlation matrix will be given as an argument to Seaborn's heatmap function.

### **Clearly describe input features and outputs for ML methods.**

There are a total of 11 columns that are divided into input features and target output columns.

* **Input features** includes Gender, Age, Race (Reported), Height (cm), Weight (kg), Diabetes, Simvastatin (Zocor), Amiodarone (Cordarone), INR on Reported Therapeutic Dose of Warfarin, VKORC1 genotype: -1639 G>A (3673); chr16:31015190; rs9923231; C/T
* **Target column** is Therapeutic Dose of Warfarin.

### **Describe the machine learning methods you choose to study the topic, including a description of algorithms.**

1. **Stacked Generalization Regression Model:** Stacked Generalization is an ensemble learning method that improves the accuracy of predictions by combining multiple base models. In this method, the predictions of multiple base models are fed into a meta-model that trains how to integrate the predictions to generate the final output.

In our project, three fundamental models are used: Linear Regression, Random Forest Regression, and Gradient Boosting Regression. A Linear Regression model becomes known as the meta-regression, which accepts as input the predictions of fundamental models and trains to combine them to generate the final output. The mlxtend library's StackingCVRegressor class is used to implement the layered generalization model. This class accepts the fundamental models and the meta-regression as input and trains the model using K-Fold cross-validation.

1. **Logistic Regression:** Logistic regression is a classification method that is implemented for predicting the probability of a binary result (i.e., two potential classes) using one or more predictor variables. In the project, the 'Therapeutic Dose of Warfarin' output variable is transformed into binary classifications: high required dose (HRD) and low required dose (LRD).
2. **Support Vector Machine:** Three variants of SVM algorithms are implemented here: SVM Classifier, SVM Regressor, and One-class SVM for outlier detection.

* *SVM Classifier:* Support Vector Machine is a method for supervised learning that can be used for tasks such as regression and classification. Here we implement the SVM classifier with the radial basis functions (RBF) kernel and 'auto' gamma value, that indicates that the gamma value is calculated automatically based on the input data. The algorithm finds the region of the hyperplane that most effectively isolates the two classes.
* *SVM Regressor*: For regression tasks, SVM regressor is implemented. The algorithm attempts to identify the hyper-plane that best suits the data points, i.e., an equation that forecasts the target variable given new input data.
* *One-class SVM:* The unsupervised learning algorithm used for identifying outliers is one-class SVM.

1. **Ensemble Learning and Random Forest:** Here, we apply an ensemble learning approach based on stacking, a technique which uses multiple models to enhance prediction accuracy. Ridge regression, Lasso regression, K-Nearest Neighbours regression, Decision Tree regression, Linear Support Vector regression, Radial Basis Function Support Vector regression, Random Forest regression, and Gradient Boosting regression are the base models used in this ensemble.

The implemented meta model is a 100-estimator Gradient Boosting regression model. The ensemble model is set up using the mlxtend library's StackingCVRegressor class, which enables us to define the base models, meta model, and cross-validation strategy. Based on the training data, the base models are fitted, and their predictions are used as attributes in the meta model.

This code generates an ensemble learning model using layering and measures its accuracy on a regression problem. Various base models and a meta model can enhance the accuracy of predictions and reduce the risk of overfitting by reducing the likelihood of overfitting.

Afterwards, the data is divided into training and testing collections, with 20% of the dataset used for testing. On the training data, the ensemble model is fitted, and on the testing data, predictions are made. Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared are calculated as evaluation metrics.

1. **Decision Tree:** The Decision Tree algorithm has been applied for classification as well as regression assignments. It is a non-parametric model that constructs a tree-like structure in which each internal node represents a feature-based decision, each branch represents the result of the decision, and each of the leaf nodes depict a class label or a continuous output result. Using the DecisionTreeClassifier class from the scikit-learn library, the provided code creates a classification model.
2. **Artificial Neural Network:** Neural networks are a category of artificial intelligence models that are based on the structure as well as the functioning of the human brain. Neurons, which are interconnected nodes, are responsible for processing and transmitting information via weighted connections.

Using the Keras library, a fundamental artificial neural network (ANN) is constructed in the provided code. Each of the network's three dense layers has an independent number of neurons and activation functions. The input layer contains 64 neurons and is activated using the Rectified Linear Unit (ReLU) function. The second hidden layer comprises 32 neurons and is activated using the ReLU activation function with an L1 regularization term of 0.001. The output layer consists of a single neuron with a linear activation function. The model is developed using an optimizer named Adam with a learning rate of 0.01 and the mean square error (MSE) as the loss function.

### **Describe how the ML model/results will be evaluated.**

Evaluation Metrics that are used for machine learning algorithms for regression and classification tasks entails assessing the correctness of the predictions made by the model on a test dataset using the appropriate performance metrics. Common metrics of assessment for regression and classification assignments include the following:

**Regression Tasks**

* *MSE* calculates the mean squared difference between the expected and actual values. A value with a reduced MSE indicates superior efficacy.
* *Root square of the MSE* is the RMSE. It gives a measure of the prediction error standard deviation.
* *MAE* determines the mean absolute difference between the actual and expected values. MSE is more sensitive to outliers than this method.
* *R-squared (R2):* It determines the proportion of the objective variable's variance that the model explains. A greater R2 value signifies greater effectiveness.

**Classification Tasks**

* *Confusion Matrix:* It assesses the proportion of the model's predictions that were accurate. The formula is (TP+TN)/(TP+TN+FP+FN).
* *Precision:* It determines the proportion of true positive instances among all predicted positive instances. It is helpful when the intent is to reduce false positives. It quantifies the proportion of genuine positives among the model's positive predictions. It is determined by TP/(TP+FP).
* *Accuracy:* It is the most fundamental and commonly used evaluation metric, which measures the proportion of instances correctly classified out of the complete number of instances in the test set.
* *Recall/Sensitivity:* It determines the proportion of true positive instances that are accurate positives. It is helpful when the objective is to reduce false negatives.
* *F1-Score:* It is the harmonic average of precision and recall and provides an individual score that is proportional to both metrics.
* *ROC Curve:* It provides an illustration of the trade-off between sensitivity and specificity by plotting the true positive rate opposed to the false positive rate at various threshold values.

## 

## **Chapter 03 - Results**

### **Model selection and Hyper parameter tuning**

It is the procedure of determining the optimal model and its related hyperparameters for a given dataset. This is a crucial phase in the application of machine learning as it improves the model's accuracy on test data. Cross-validation, a method to evaluate the accuracy of a machine learning model by training and testing it on various subsets of the data, is one means to accomplish this.

In the project, the ***GridSearchCV*** *method* from scikit-learn is applied to tune the KerasRegressor model's hyperparameters. GridSearchCV conducts a comprehensive search over the parameter grid specified by the param\_grid variable. In this instance, the parameter grid contains two hyperparameters with a pair of options each: batch\_size and epochs.

***Cross-Validation*** *is used*  to determine the accuracy of each hyperparameter combination. The data is divided into K-folds using cross-validation, and each hyperparameter combination undergoes training on K-1 folds and is evaluated on the remaining fold. This procedure repeats itself K times so that each fold is evaluated K times, yielding K evaluation scores. The highest possible score is the mean of the K scores.

### **Evaluation of Regression Models**

Table 01 shows that the Stacked Generalization Regression Model has the lowest RMSE at 1.021, observed closely by the Random Forest method at 1.038. With an RMSE value of 1.059, the ANN method has the largest average difference between predicted and actual values.

The Random Forest approach has the lowest value for the MAE metric at 0.795, which is followed by the ANN method at 0.815 and the Stacked Generalization Regression Model at 1.021.

The R2 values indicate that the ANN method performs the best with an estimate of 0.815, followed by the Random Forest method with a value of 0.780 and the Stacked Generalization Regression Model with a value of 0.780.

In general, the results indicate that Random Forest and Stacked Generalization Regression Model perform better in terms of RMSE and MAE, whereas the ANN method performs more effectively in terms of R2. It is essential to observe that the efficacy of each method may vary depending on the evaluated dataset and problem.

|  |  |  |  |
| --- | --- | --- | --- |
| **Methods** | **RMSE** | **MAE** | **R2** |
| **Ensemble Learning and Random Forest** | 1.038 | 0.795 | 44.40 |
| **Stacked Generalization Regression Model** | 1.021 | 0.780 | 46.11 |
| **Artificial Neural Network** | 1.059 | 0.815 | 42.05 |

**Table 01 - Comparing the prediction accuracy of Different Models**

### **Evaluation of Classification Models**

Table 02 demonstrates the evaluation metrics for three classification methods: Logistic Regression, Decision Tree, and SVM Classifier.

* *Accuracy:* The model with the highest accuracy is the logistic regression model, with 76.16%, followed by the SVM Classifier with 70.53% and the Decision Tree with 64.8%.
* *Specificity*: The model with the highest specificity is the logistic regression model, with 73.85%, followed by the SVM Classifier with 67.85% and the Decision Tree with 58.82%.
* *Sensitivity:* The logistic regression model has the maximum sensitivity (77.63%), followed by the SVM Classifier (72.04%) and the Decision Tree (69.0%).
* *F1 score:* The logistic regression model has the best F1 score, at 70.63 percent, followed by the SVM Classifier with 62.42 percent and the Decision Tree with 57.9 percent.
* *Precision:* The logistic regression model has the most accuracy at 73.85%, followed by the SVM Classifier at 71.42% and the Decision Tree at 58.88%.

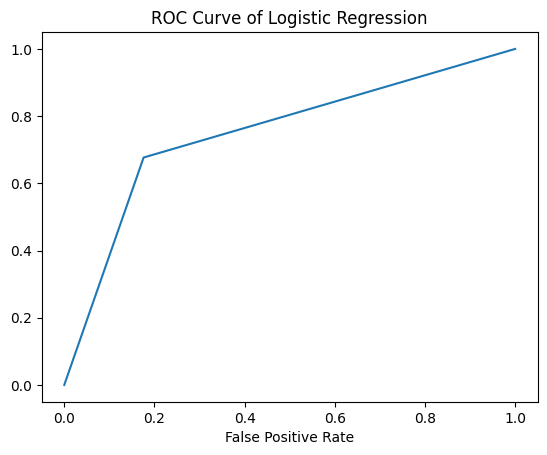
In terms of accuracy, specificity, sensitivity, F1 score, and precision, the logistic regression model performs the best overall. Except for precision, the SVM Classifier outperforms the Decision Tree in all performance metrics. Hence, the Decision Tree model has the poorest performance of the three models.

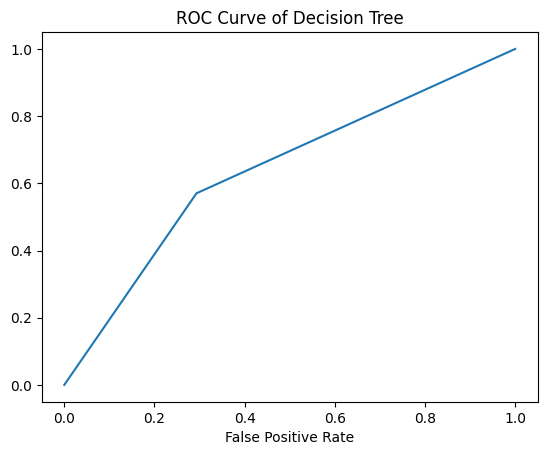
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Methods** | **Accuracy** | **Specificity** | **Sensitivity** | **Recall** | **F1 score** | **Precision** |
| **Logistic Regression** | 76.16% | 73.85% | 77.63% | 67.68% | 70.63% | 73.85% |
| **Decision Tree** | 64.89% | 58.82% | 69.12% | 57.03% | 57.91% | 58.82% |
| **SVM Classifier** | 70.53% | 67.85% | 72.04% | 57.79% | 62.42% | 71.42% |

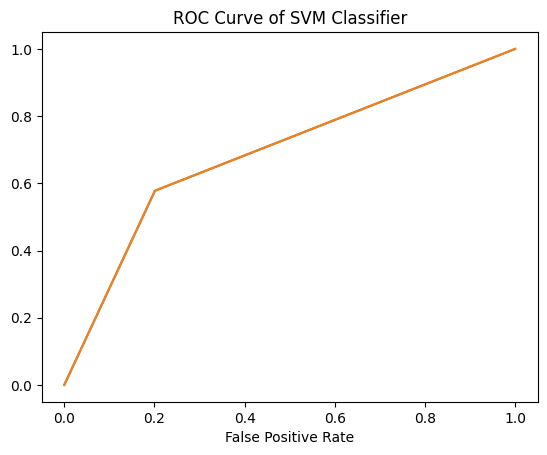
**ROC Curve of different Classification Algorithms**

The ROC (Receiver Operating Characteristics) curve is a visual illustration of a classification model's accuracy that illustrates the difference between the true positive rate (TPR) and the false positive rate (FPR) for various classification thresholds. It aids in finding the optimal classification threshold based on the performance of the model.

Using the sklearn.metrics module's roc\_curve function, it is possible to show the ROC curve of a classification model in Python.

****

****

****

A curve that is closer to the upper-left corner of the plot suggests a model with greater performance. The graph shows that the logistic regression model performs the best among the other classification algorithms.

### **How to build web applications for your ML models?**

Gradio is a Python library that facilitates the creation of shareable, customizable web-based user interfaces for machine learning algorithms. Here I train the linear regression model and use that model to make prediction of Warfarin dose for patients. The web page I created for my Linear Regression Model is shown below.

A screenshot of a computer

Description automatically generated with medium confidence

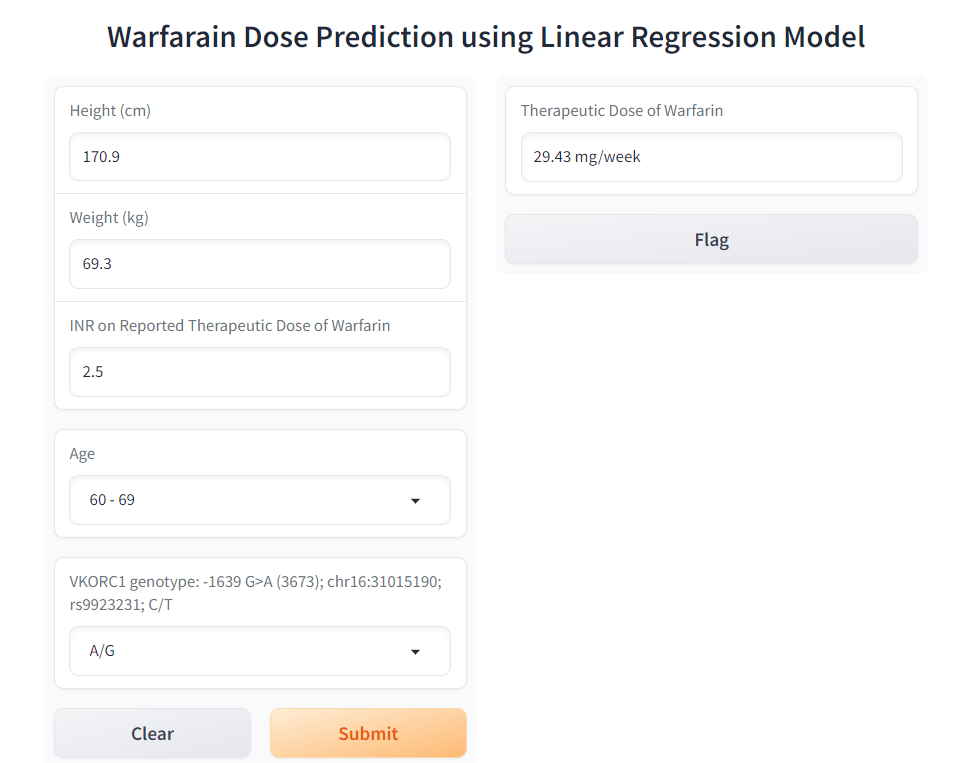
**Dropdown input options**

I followed these steps to create a website for ML model using Gradio:

* **Install Gradio:** You can install Gradio via the pip command pip install gradio to install Gradio.
* **Import your trained model:** You must import your trained machine learning (ML) model using Python's pickle module and ensure that it is accessible through your Python environment.
* **Specify the input as well as the output elements.** Use Gradio's input and outputs classes to create the web interface's input and output components. Depending on the sort of input your model requires, input components may include *Number(Height, weight and INR on Reported Therapeutic Dose of Warfarin) dropdowns( for Age and VKORC1 genotype: -1639 G>A (3673); chr16:31015190; rs9923231; C/T) and TextBox for predicting Therapeutic Dose of Warfarin.* The output elements may include text boxes, images, or elements of HTML.
* **Specify the Prediction function:** Create a function that accepts input data from the web interface, prepares it if necessary, and sends it to your trained machine learning model for prediction. The function must return the prediction as a string or in some other format that can be displayed by the output component.
* **Gradio's Interface class** is used to construct the web interface, with the input and output components and the prediction function as parameters. Customize the appearance and operation of the interface as desired.
* **Start the user interface:** Apply the launch() method of the Gradio interface object to open the web interface in a web browser. Share the URL to allow others to access your model via the web interface.

With these methods, you can develop a web application for your ML model using Gradio without the need for web development tools or frameworks.

After entering the input values, the web page loads our trained model and makes predictions on *Therapeutic Dose of Warfarin* as shown in the figure below.



**Conclusion**

Based on the analysis of the Warfarin dose prediction models, it can be discovered that logistic regression is the most accurate, specific, and sensitive classification model, with the highest F1 score. The findings of this study indicate that ensemble methods of learning, such as Random Forest and Stacked Generalized Regression Model, can be effective for predicting the Warfarin dose, whereas logistic regression may be the optimal choice for classification tasks. However, additional research and analysis are required to determine the optimal model for a particular dataset and problem.

**Future Work**

The future work can involve investigating the efficacy of methods that have not been examined in the present research. The study can be enlarged by investigating the most effective predictors for patients with ethnicities. Additionally, optimizing the models' hyperparameters can improve their efficacy. Finally, the models can be applied to additional datasets to evaluate their generalizability.