Final Project: Warfarin Dose Prediction

Ethan Kao, Zeqing Zhu

Saint Louis University

December 16, 2022

**Abstract**

The effectiveness of warfarin, a commonly prescribed anticoagulant, is highly dependent on the individual patient's response to the drug. Determining the proper dose of warfarin for a patient is a complex task that requires careful monitoring and adjustment. In this study, we compared the performance of several machine learning models for predicting warfarin dose. We evaluated the K-nearest neighbors (KNN), random forest, support vector classification, logistic regression, voting classifier, and neural network models using a dataset of patient information. The results showed that the neural network model performed best, achieving 81% accuracy in predicting warfarin dose. The performance of other models is similar with an accuracy rate of about 80%, and the KNN and random forest models are slightly worse with an accuracy rate of about 79%. These results demonstrate the potential of machine learning to improve individualization of warfarin dosing.

**Introduction**

Warfarin is a medication that is commonly used to help prevent blood clots from forming in the body. It is an effective medication, but it requires careful dosing because the right amount can vary greatly from person to person. Taking too much warfarin can lead to serious complications, such as bleeding, while taking too little can result in blood clots forming. One way to determine the right dose of warfarin for a person is to use a method called "dose optimization," which involves a healthcare provider taking several factors into account, such as the person's age, weight, and medical history, to determine the best starting dose (Ruth Jessen Hickman, 2022). However, this process can be time-consuming and is not always accurate. This is where machine learning comes in. Machine learning is a type of artificial intelligence that involves training algorithms on large amounts of data to enable them to make predictions or take actions without being explicitly programmed to do so. In the case of warfarin dosing, machine learning algorithms can be trained on data from previous patients who have taken warfarin, such as their age, weight, medical history, and warfarin dose. Using this data, the algorithm can learn to predict the optimal warfarin dose for a new patient based on their individual characteristics. This can help healthcare providers determine the right warfarin dose for a patient more quickly and accurately than using traditional methods.

In addition to improving the accuracy and speed of warfarin dosing, using machine learning can also help to reduce the risk of complications associated with the medication. By accurately predicting the optimal warfarin dose for a patient, healthcare providers can help to prevent the patient from taking too much or too little of the medication, which can help to reduce the risk of bleeding or blood clots. Overall, by training algorithms on large amounts of data, healthcare providers can more easily determine the right warfarin dose for a patient, which can ultimately lead to better patient outcomes.

In this paper, we will explore how different machine learning models can be used to predict the correct Warfarin dose for a given patient. Specifically, we will discuss how KNN, Random Forest, Support Vector Classification, Logistic Regression, VotingClassifier, and Neural network models can be used for this purpose. We will also provide some examples of how these models can be implemented in Python.

**Method**

**Pipeline:**

In this project, we collect a dataset from the PharmGKB website (<https://www.pharmgkb.org/downloads>). After inspecting the dataset and reviewing project requirements, data preprocessing was applied. Following, we trained several Models: KNeighborsClassifier, Random Forest, Support Vector Classification, Logistic Regression, VotingClassifier & Neural Network, and tuning was applied to each classifier. Last, we deploy our results to Hugging Face, keep monitor and maintaining the system (Fig. 1).

****

Fig.1 Pipeline of Final Project

**Preprocessing:**

1. Remove unnecessary columns

We trimmed unnecessary columns in excel. After, loading the modified file into google colab (Table 1), we got 12 features and 1 target (Therapeutic Dose of Warfarin) in this file.

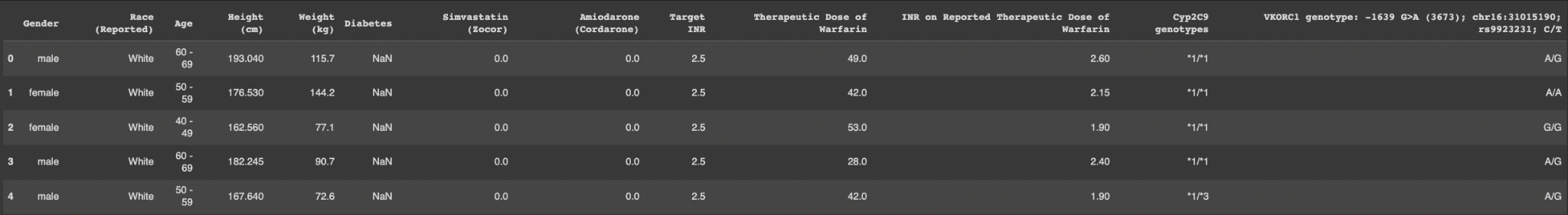
****

Table .1 Modified file, 5700 samples were stored. The top five samples were displayed in this table.

1. Inspect the data

In this file review, we notice 5700 samples were stored, and each feature has N/A value, need to apply preprocessing before training (Fig .2).

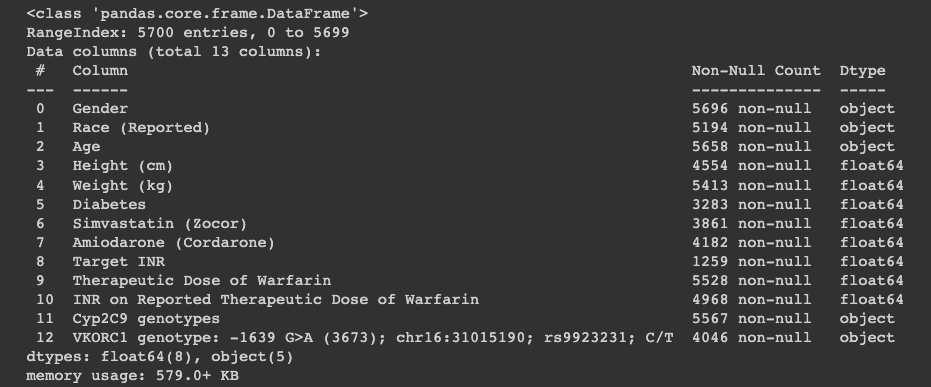
****

Fig .2 Inspect the Modified file

1. N/A value trimming

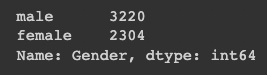
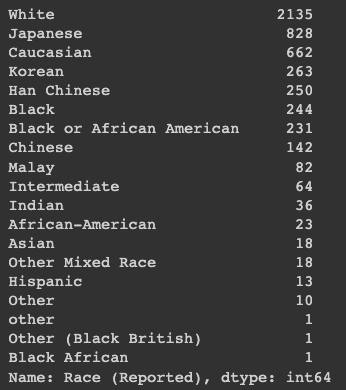
First, we trimmed the dataset based on the value missing in the target (Therapeutic Dose of Warfarin)(Sample decrease: 5700 → 5528). After, we noticed only 4 Gender values were missing, and in (Fig .3) we found out male was the majority group so we replaced the N/A value with male. (Fig .4) Shows white was the majority group so we replace the N/A value with white. For the age feature, there were only 39 missing values so we decided to drop the sample with missing age(Sample decrease: 5528 → 5489).

For the height and weight feature, we decided to fill the N/A value with the mean value of all data.

For Diabetes, Simvastatin, Amiodarone, and Target INR we decided to fill the N/A value with 0.

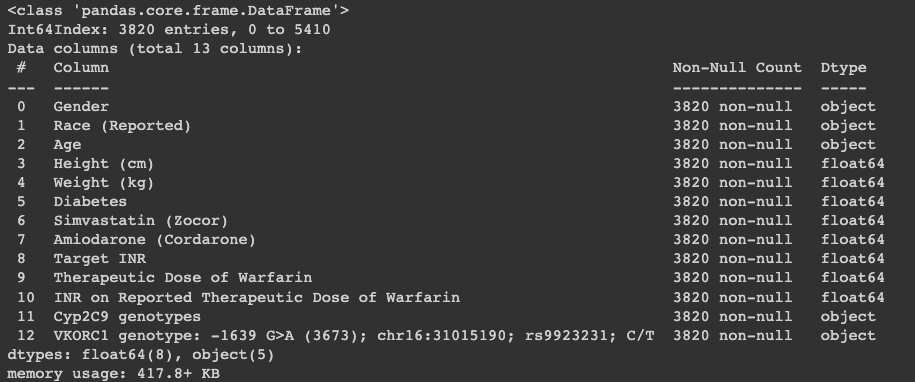
For INR on Reported Therapeutic Dose of Warfarin, we decided to fill the N/A value with the avg value of all data.

For Cyp2C9 genotypes and VKORC1 genotypes, we don’t know how these two features will affect the model so we decided to drop the N/A value.





After N/A value trimming 3820 samples survived (Fig .5).

 Fig .5 Inspect file after N/A trimming

1. Categorical feature transformation

In this step, we apply sklearn.preprocessing.LabelEncoder to transform categorical features to continuous number labels (Table .2). After, each feature was changed to the same type using astype function (float64).

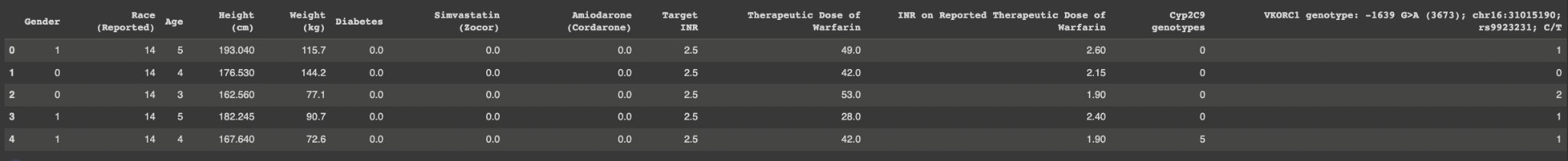
****

Table .2Applied LabelEncoder to transform categorical features to continuous number labels.

1. Train/test split and convert multi-class label to binary class

We split the dataset into 70% training and 30% testing using sklearn.model\_selection.train\_test\_split, and set the random state to 42.

Based on project requirements, we split required doses >30 mg/wk (high required dose (HRD)) into one class and the second class contains the patients who need doses of ≤30 mg/wk (low required dose (LRD)).

1. Histogram of features and target

After visualizing the feature and target we notice the dataset is imbalanced, so we decided to apply standardization to features increasing accuracy when training the model (Fig. 6).

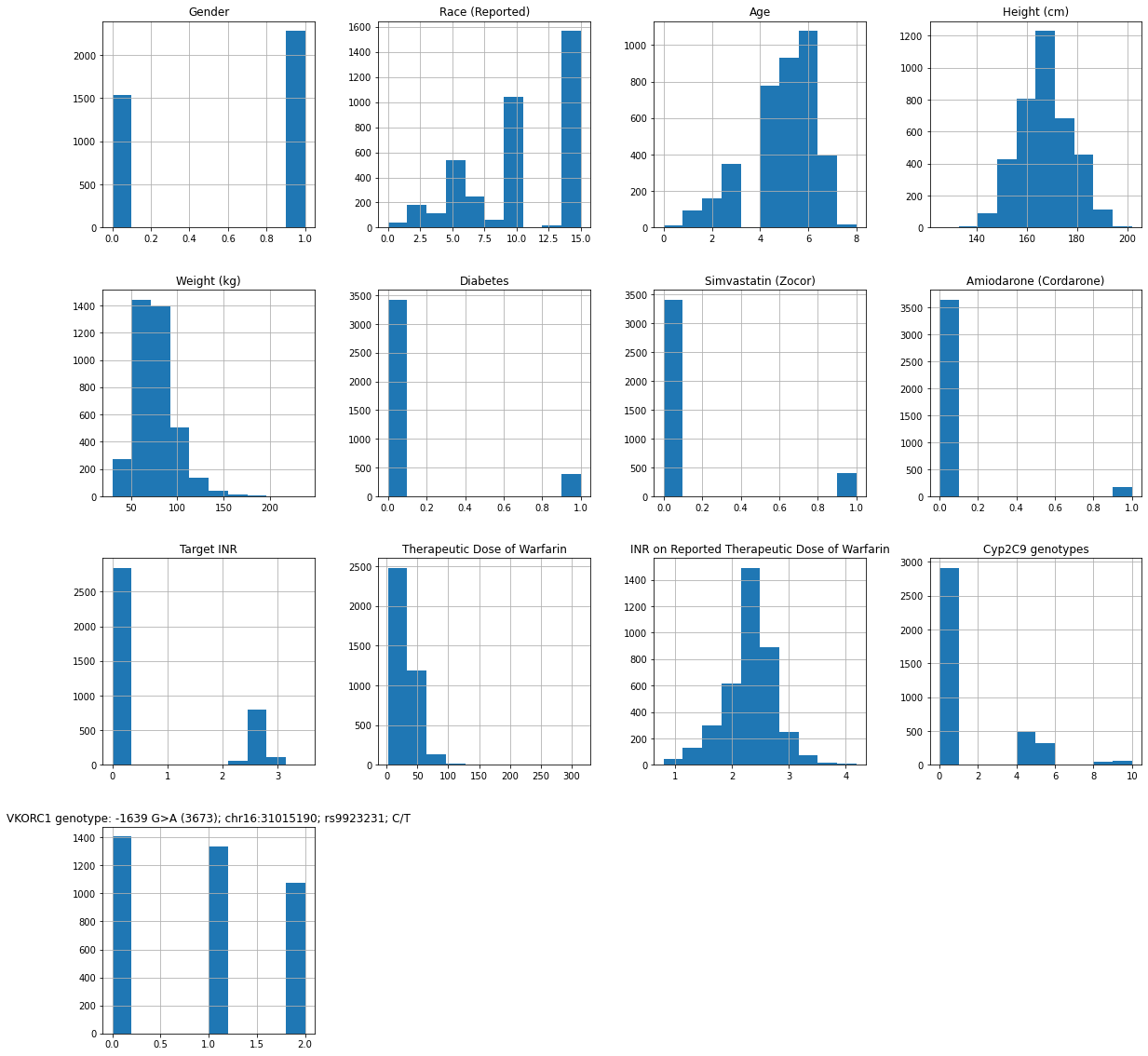
****

Fig .6Histogram of features and target

1. Standardization

sklearn.preprocessing.StandardScaler was applied to X\_train\_feature and X\_test\_feature to increase accuracy when training the model.

**Model training:**

After data preprocessing and standardization, we trained several models using different classifiers.

1. KNeighborsClassifier (KNN)

KNeighborsClassifier is a supervised learning algorithm that can be used for both classification and regression tasks. The algorithm works by calculating the distance between a new sample and the training samples. It then finds the K samples in the training data that are closest to the new sample, based on this distance measure. The class label of the new sample is then predicted based on the majority class label of the K nearest neighbors (Harrison, 2019).

The number of nearest neighbors, K, is a user-defined hyperparameter of the model. A larger value of K will result in a smoother decision boundary and a more conservative model, while a smaller value of K will produce a more complex decision boundary and a potentially overfit model.

One of the main advantages of KNeighborsClassifier is its simplicity and flexibility. It is easy to implement and can be used with a wide variety of distance measures, allowing it to be applied to a wide range of data types and problem domains. However, one of the main disadvantages of this algorithm is that it can be computationally expensive to find the nearest neighbors for a large dataset.

1. Random Forest (RF)

Random Forest is an ensemble learning method for classification and regression tasks. It is a type of supervised machine learning algorithm that works by building multiple decision trees, which are then combined to make a more accurate and stable prediction (Li et al., 2021).

The individual decision trees in a Random Forest are trained using a bootstrapped sample of the training data. This means that each tree is trained on a different subset of the training data, with replacement. This allows each tree to make predictions based on different subsets of the features, leading to a diverse set of trees that can capture different aspects of the data.

One of the main advantages of Random Forest is its ability to handle large and complex datasets and to make accurate predictions. It also has relatively little need for hyperparameter tuning, as the randomness of the algorithm helps to avoid overfitting. However, one disadvantage of Random Forest is that it can be slow to make predictions, as it must make a prediction using each individual tree.

1. Support Vector Classification (SVC)

Support Vector Classification (SVC) is a linear classification algorithm used to classify data into two or more classes. It is a supervised learning algorithm that seeks to find a hyperplane that maximally separates the classes in the training data. The hyperplane is chosen such that it maximizes the margin, or distance, between the closest points of the two classes (Gandhi, 2018).

The algorithm works by mapping the input data into a high-dimensional feature space, where it is then possible to find a separating hyperplane. The training data is used to find the optimal hyperplane, and new samples are then classified based on which side of the hyperplane they fall on.

One of the main advantages of SVC is that it can achieve good performance even when the data is not linearly separable. This is because the algorithm can use the kernel trick to map the data into a higher-dimensional space, where a linear separation may be possible. However, one disadvantage of SVC is that it can be computationally expensive to train, especially for large datasets.

1. Logistic Regression (LOG)

Logistic Regression is a classification algorithm used to predict a binary outcome. It is a supervised learning algorithm that uses a linear combination of the input features to make predictions. The predictions are then transformed using the logistic function, which maps the output to a value between 0 and 1, which can be interpreted as the probability that the given sample belongs to the positive class(Kanade, 2022).

The Logistic Regression algorithm works by finding the optimal coefficients for the input features, such that the predicted probabilities are as close as possible to the true class labels. This is done using an optimization algorithm, such as gradient descent, to minimize the loss function. The loss function measures the difference between the predicted probabilities and the true class labels.

One of the main advantages of logistic regression is its simplicity and interpretability. The coefficients of the input features can be used to understand the relative importance of each feature in predicting the outcome. However, one disadvantage of Logistic Regression is that it is limited to binary classification tasks, and it cannot handle non-linear relationships between the input features and the output.

1. VotingClassifier

VotingClassifier is a machine learning ensemble method that combines the predictions of multiple individual classifiers to make a final prediction. It is a simple and effective way to improve the performance of a classifier by leveraging the strengths of multiple models.The basic idea behind VotingClassifier is to train multiple classifiers on the same dataset, then have each classifier make predictions on a new input. The final prediction is made by taking the majority vote or by taking the average of the predictions of the individual classifiers. This approach can help reduce the variance and bias of the overall model, leading to improved performance (Nair, 2021).

VotingClassifier can be used with any type of classifier, including decision trees, support vector machines, and neural networks. It is particularly useful when the individual classifiers are trained on different subsets of the data or using different algorithms. In this case, each classifier may make predictions that are different from the others, leading to a more diverse set of predictions and a more accurate final prediction.

We pass the best parameter from LogisticRegression, RandomForest, SVC, KNN classifier to make a hard voting classifier

1. Neural network

Neural networks are a type of machine learning algorithm modeled after the structure and function of biological neural networks. They consist of layers of interconnected artificial neurons that receive input, process it, and produce an output. Each layer consists of multiple nodes, and the number of layers and nodes can vary depending on the complexity of the problem being solved. The nodes in the input layer receive data from the outside world and pass it through to the other layers. The hidden layers then process this information and identify patterns and relationships within the data. Finally, the output layer produces a desired result based on the patterns identified in the data (IBM Cloud Education, 2020).

The power of neural networks lies in their ability to learn from data. Through a process known as backpropagation, neural networks can adjust the weights and connections between neurons to continually refine their structure and improve their performance over time. This enables them to handle complex tasks such as natural language processing, image recognition, and speech recognition. By incorporating additional tricks such as regularization, dropout, and batch normalization, neural networks can further increase their accuracy and speed.

Neural networks have become a crucial tool in many areas of artificial intelligence and machine learning. By training large datasets and leveraging advanced modeling techniques, they offer unprecedented levels of accuracy and performance compared to traditional algorithms. With their ability to quickly process large amounts of data and their high level of accuracy, neural networks will continue to play a major role in shaping the future of AI technology**.**

**Results**

**Model tuning & selection:**

We applied sklearn.model\_selection.RandomizedSearchCV to find the best parameter for each classifier. Accuracy, precision, recall, AUC, and F1 score were calculated.

* Traditional ML algorithm:

1. KNN

We pass {n\_neighbors\_list = [3, 5, 7,9,11], metrics\_list = [distance.euclidean, distance.minkowski, distance.cosine]} for RandomizedSearchCV because the run time was too long so we only run (n\_iter = 10, cv = 5), and get the best parameter {'n\_neighbors': 11, 'metric': cosine} (Fig. 7).

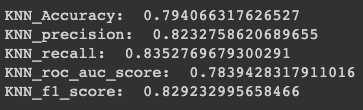


Fig .**7** KNN score report

1. RandomForest

We pass {'n\_estimators': [100, 200, 300, 400], 'max\_features': ['auto', 'sqrt'],"min\_samples\_split":[2,3,4,5],"min\_samples\_leaf":[1,2,3,4]} for RandomizedSearchCV(n\_iter = 10, cv = 10), and get the best parameter {'n\_estimators': 100, 'min\_samples\_split': 4, 'min\_samples\_leaf': 3, 'max\_features': 'sqrt'} (Fig .8).

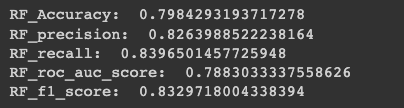


Fig .8 RF score report

1. SVC

We pass {'C': [0.001, 0.01, 0.1, 1,3,5,10], 'kernel': ['linear',"poly", 'rbf'], 'gamma': ["auto","scale",0.001, 0.01, 0.1, 1]} for RandomizedSearchCV(n\_iter = 10, cv = 10), and get the best parameter {'kernel': 'linear', 'gamma': 'auto', 'C': 5} (Fig .9).

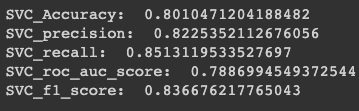


Fig .9 SVC score report

1. Logistic Regression

We pass {'C': [1,0.1,0.01,0.001,0.0001,0]} for RandomizedSearchCV(n\_iter = 10, cv = 10), and get the best parameter is C=0.1 (Fig .10).

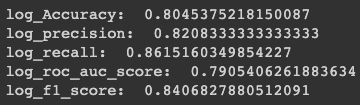


Fig .10 Logistic Regression score report

1. VotingClassifier

We pass the best parameter in each above classifier and build a hard voting classifier (Fig .11, 12).

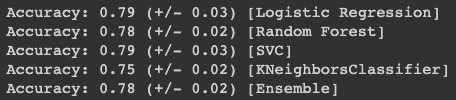


Fig .11 Classifiers score report in voting classifier

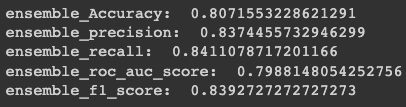


Fig .12 Voting classifier score report

In this figure we can observe that LOG, RF, SVC, and Ensemble classifier have similar performance and they are slightly better than KNN (Fig. 13).

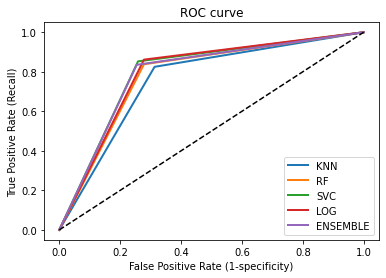


Fig. 13 AUROC of every traditional classifier

* Deep learning ML algorithm:

1. Neural Network

First, we build a 3-layer sequential model, one input layer, one hidden layer, and one output layer. We fit the model with 0.3 validation split and 100 epochs. We found the model is overfitting when epoch > 20 (Fig .14), so we decrease the epochs to 20 and apply learning rate tuning(Fig .15).

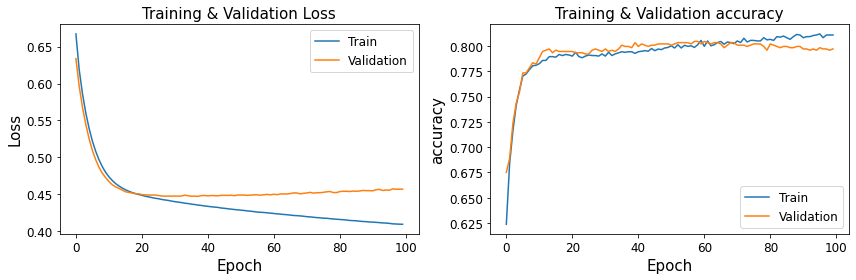


Fig .14 Loss & accuracy score under 100 epochs

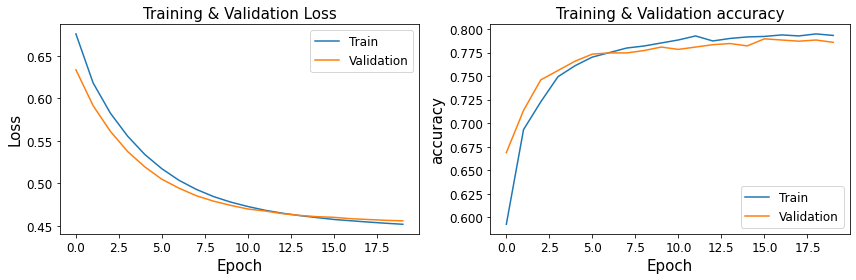


Fig .15 Loss & accuracy score under 20 epochs

We first tuned the learning rate with parameter = [0.001, 0.01, 0.1], and found out that learning rate = 0.1 have the highest accuracy and lowest loss (Fig .16).

After, we pass the learning rate condition and apply layer tuning, layers = [1,2,3,4,5]. We found layer = 3 and learning rate = 0.1 have the highest accuracy and lowest loss (Fig .17).

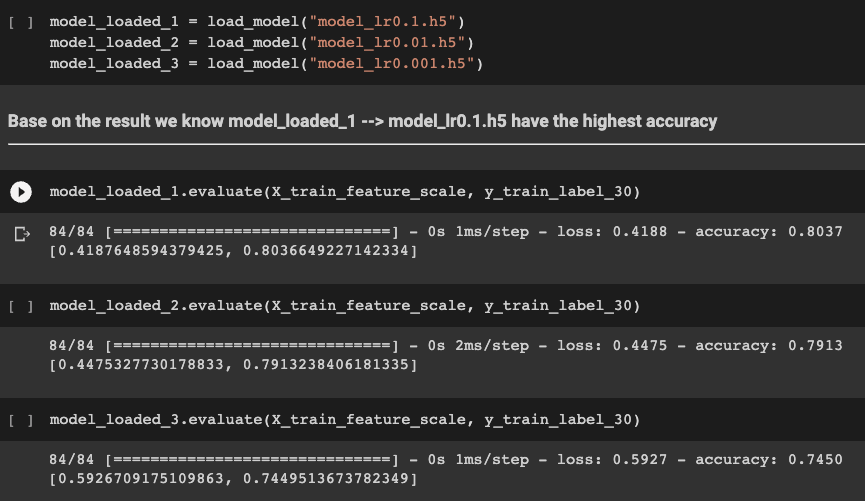


Fig .16 Deep learning ML learning rate tuning

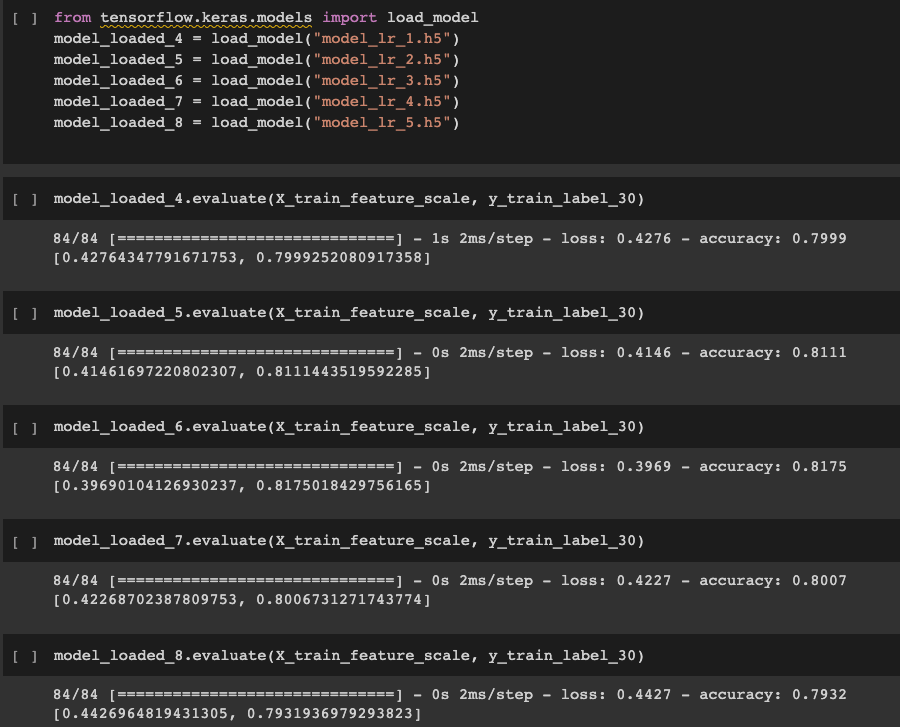


Fig .17 Deep learning ML layer tuning

**Possible future work:**

We can improve the preprocessing part. For now, we drop multiple genotype columns from the raw dataset and drop the sample when we have N/A value in it, with more biological background we can treat the data more precisely and get a larger dataset when training the model, also we can build a regression model to get more accurate dose suggestion.

**Contribution**

1. Structure: We have several meetings to build the pipeline together.
2. Model training:We both build our own version of the model (preprocessing strategy), and finally come up with 5 traditional models and 1 deep learning model.
3. Model deployment:For final deployment, we decided to use Ethan’s preprocessing strategy and Zeqing contribute to Gradio and Hugging face.

<https://huggingface.co/spaces/ZQ12/WarfarinDosePrediction>

**Course Summary**

By the end of the machine learning course, we have a solid understanding of the principles and techniques behind modern machine learning methods. Over the course of this semester, we learned about supervised learning, including linear and nonlinear regression, support vector machines, and neural networks, as well as unsupervised learning using deep learning, ensemble methods, probabilistic models, clustering, and dimensionality reduction. Furthermore, after extensive practice, we are able to effectively implement these methods to solve real-world problems and design and optimize models for complex tasks. Armed with this knowledge and experience, in our opinion, students who join this class will be equipped to delve into the field and make valuable contributions to the development of artificial intelligence.

**Reference**

Gandhi, R. (2018, July 5). *Support Vector Machine - introduction to machine learning algorithms*. Medium. Retrieved December 14, 2022, from <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

Harrison, O. (2019, July 14). *Machine learning basics with the K-nearest neighbors algorithm*. Medium. Retrieved December 14, 2022, from <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

IBM Cloud Education. (2020, August 17). *What are neural networks?* IBM. Retrieved December 14, 2022, from <https://www.ibm.com/cloud/learn/neural-networks>

Kanade, V. (2022, April 18). *What is logistic regression? equation, assumptions, types, and best practices*. Logistic Regression: Equation, Assumptions, Types, and Best Practices. Retrieved December 13, 2022, from <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/>

Li, G.-Q. *et al.* (2021) *Application of immune infiltration signature and machine learning model in the differential diagnosis and prognosis of bone-related malignancies*, *Frontiers in cell and developmental biology*. U.S. National Library of Medicine. Available at: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8082117/#:~:text=Random%20forest%20is%20an%20ensemble,the%20class%20of%20each%20tree>

Nair, A. (2021, October 12). *Combine your machine learning models with voting*. Medium. Retrieved December 14, 2022, from <https://towardsdatascience.com/combine-your-machine-learning-models-with-voting-fa1b42790d84>

Ruth Jessen Hickman, M. D. (2022, November 3). *Warfarin: Purpose, side effects, and management*. Verywell Health. Retrieved December 14, 2022, from <https://www.verywellhealth.com/what-to-know-about-warfarin-4178509>