SCHOLARLY REVIEW ONLINE

Scholarly Review Online - Fall 2024

Machine Learning as a Tool for Auxiliary Diagnosis of Osteoporosis

By Yixi Heng

AUTHOR BIO

Isabelle Heng, a senior at United World College Changshu, China, is dedicated to computer science, mathematics, and Artificial Intelligence, having learned programming languages including Python, JavaScript, HTML, C, and C++, and is devoted to robotics, web development, app development, algorithms, and machine learning; with a steadfast commitment to her passion, Isabelle strives to continue advancing her knowledge and deepening her insights into computer science and its interdisciplinary applications, including LLM and AI audio generation models with emotional values, as well as interactive robotics devices like surgical robots and pilotless automobiles, recognizing the change-making potential and problem-solving power of computer science to create positive impacts on the world through technology.

ABSTRACT

By making predictions from the model created with machine learning, specifically supervised learning, this essay focuses on the auxiliary diagnosis of osteoporosis. The chosen dataset includes highly relevant indexes that determine the diagnosis of osteoporosis. After doing the feature selection and one-hot encoding to preprocess the dataset, we then split the dataset into validation and training sets to avoid overfitting. Then, to train the model for prediction, we used the neuron network, chose ReLU and Sigmoid functions as activation functions, and changed the number of neurons and epochs to find the best-fit model. Through further measurement, evaluation, and application, including plotting the binary cross entropy error diagram and analyzing the classification report, the model is examined as highly accurate and desirable. With an appreciable degree of accuracy, this model for auxiliary diagnosis is able to prevent disease development or deterioration by detecting early symptoms and vulnerable populations, reduce the probability for doctors to misdiagnose, and improve the efficiency and quality of the osteoporosis diagnosis, therefore enhancing patient experiences.

Keywords: machine learning; supervised learning; osteoporosis

BACKGROUND AND INTRODUCTION

Osteoporosis is a skeleton disorder in which the symptoms include bones becoming weak and brittle and being more likely to break the bone. Although it occurs among all races and genders, non-Hispanic white and Asian women, especially older women after menopause, are at the highest risk, "Osteoporosis is 4 times more common in women than men"((Bailey, 2011). Besides, family history and other health indexes are also factors when assessing the disease. It is hard to perceive and not often paid attention until the symptoms become evident till a point when they must see the doctor. Hence, auxiliary diagnosis and preventive treatment of the disease is crucial.

Machine learning is a subfield of artificial intelligence that uses data and algorithms to develop computational models, enabling computers to extract patterns and learn. In application, machine learning serves as a powerful and effective method to make predictions and help make decisions.
This essay applies supervised learning, a branch of machine learning, to build a model for predicting the risk of osteoporosis as a tool for auxiliary diagnosis. Data in supervised learning is composed of a set of examples and the data for each example is a set of features and a label, which is a targeted variable. Supervised learning aims to develop a model, or in other words, a function to predict a label for features in a newly given example.

DATA SET INTRODUCTION

`To achieve the goal of auxiliary diagnosis at a relatively high accuracy, choosing relevant and authentic data is a fundamental step. The data set used in the research is chosen from the website Kaggle (Amit Kulkarni, 2024). Comprised of 1958 examples, 15 features and 1 label indicating whether the person is predicted to be diagnosed with osteoporosis or not for each example, these traits of the dataset suggest that our method of building the prediction model is supervised learning. The table below shows the head of the raw data.

Table 1. Sample of the raw dataset

The output label of each example is either 1, diagnosed with osteoporosis, or 0, not diagnosed with osteoporosis, and so binary classification can be used.

	Id	Age	Gender	Hormonal Changes	Family History	Race/ Ethnicity	Body Weight	Calcium Intake	Vitamin D Intake	Physical Activity	Smoking	Alcohol Consumption	Medical Conditions	Medications	Prior Fractures	Osteoporosis
1																
2	104866	69	Female	Normal	Yes	Asian	Underweight	Low	Sufficient	Sedentary	Yes	Moderate	Rheumatoid Arthritis	Corticosteroids	Yes	1
3	101999	32	Female	Normal	Yes	Asian	Underweight	Low	Sufficient	Sedentary	No	None	None	None	Yes	1
4	106567	89	Female	Postmenopausal	No	Caucasian	Normal	Adequate	Sufficient	Active	No	Moderate	Hyperthyroidism	Corticosteroids	No	1
5	102316	78	Female	Normal	No	Caucasian	Underweight	Adequate	Insufficient	Sedentary	Yes	None	Rheumatoid Arthritis	Corticosteroids	No	1
6	101944	38	Male	Postmenopausal	Yes	African American	Normal	Low	Sufficient	Active	Yes	None	Rheumatoid Arthritis	None	Yes	1
7	102265	41	Male	Normal	Yes	Caucasian	Normal	Low	Sufficient	Active	Yes	Moderate	Rheumatoid Arthritis	Corticosteroids	Yes	1

DATA PREPROCESSING

The data preprocessing process includes checking the categories; evaluating the data balance; feature engineering which transforms, creates, or deletes some certain features before machine learning techniques are applied; numeralizing the categories if necessary; and scaling the categorical features.

The first thing noticed is the feature, 'ID', is not affecting the osteoporosis prediction. Hence, the 'ID' column is removed. Secondly, it is important to evaluate the data balance. We need to check the number of times each category appears in each feature; for example, if a category only appears once among thousands of examples, we should not consider this example since it is not representative. We first find out how many unique categories each feature has. We then calculate the number of times each category appears. After this, we check how many examples have the label of being diagnosed with osteoporosis (1) and how many have the opposite (0). The dataset is balanced, and so there is no need for additional changes. The last step of preprocessing of the data is one-hot encoding, a technique used to represent categorical data in a numerical format. One-hot encoding converts each category into a binary vector where now each element represents a unique category. In the vector, only one element is "hot" (1) while all others are "cold" (0). Thus, in our dataset, the original feature "Medical Conditions" with categories "Rheumatoid Arthritis", "None", and "Hyperthyroidism" now becomes three features (Rheumatoid Arthritis, None, Hyperthyroidism), and for every example, the categories for these three features are either $[0, 0, 1]$, $[0, 1, 0]$, or $[1, 0, 0]$. After one-hot encoding, all the categories are numeralized, facilitating our further data processing and model construction.

Isolation Forest, an algorithm for data anomaly detection (Fei Tony Liu, 2008), is adopted to detect and remove the outliers in the dataset. After using this algorithm, the shape of the dataset stays the same as before removing, indicating no outlier.

TRAINING AND VALIDATION SETS

When modeling data from a set, the aim is to choose the most accurate and best-fit model with minimal errors. However, overfitting may occur where a model performs extremely well on the current dataset but fails to generalize to new, unseen data. In other words, the model becomes too specialized only to this dataset and loses its ability to make accurate predictions on new ones. This can lead to poor performance and inaccurate predictions when the model is applied in practical uses, causing mistakes in diagnosis.
To solve this problem, an effective method is splitting the data into a training set and a validation set, by a specific proportion. Later when training the model, only the training set will be used while the validation set is for measuring the accuracy. That is, we will compute the error based on the validation set.

In this case, we separate the dataset into a training set and a validation set with a test size of 0.25, meaning that data in the training set accounts for one-fourth of the total data; and the random state is equal to 4, to ensure that each partition result is the same, easy to debug and reproduce.

Since all the categories are binary values, scaling the features is not needed. Thus, after splitting the data, we can start training the model.

BINARY CROSS-ENTROPY ERROR

To examine the difference between predicted outputs produced by the model and the actual value, the loss function is used. The loss function for the binary classification problem is referred to as the binary cross entropy error. It measures the classification model's performance, estimating the likelihood that a given sample belongs to the positive class (target class that we are specifically interested in identifying). The loss J, therefore, is very small when the predicted and actual values are close, and vice versa.

Denote y as the label of one of the n sets

of examples, where $y = 0\ or\ 1$. Denote $\hat{y}$ as the predicted value of label y, where $0 < \hat{y} < 1$. The prediction of the model is that the example belongs to the positive class if $\hat{y}$>0.5, and the negative class if $\hat{y} < 0.5$.

Notice that in the logarithm function as shown in Figure 1, when the x value is close to 1, the y value is arbitrarily small. If the actual label is 1 but the predicted value is very low, the loss can be expressed as $y \times log(\hat{y})$, where $log(\hat{y})$ approaches 1, so the loss $y \times log(\hat{y})$ approaches 1 as well. The logarithm of $\hat{y}$ ensures that the loss increases exponentially as the predicted value approaches 0.

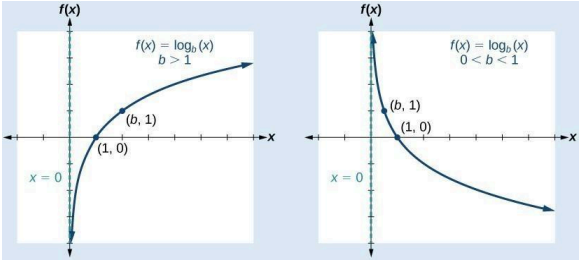

Figure 1. Logarithm function

In the other case, when the actual label is 0 but the predicted value is very high, we use $(1 - y) \times log(1 - p)$, where the absolute value of $log(1 - p)$ is very big (it is either positively or negatively approaching infinity). The logarithm of $(1 - \hat{y})$ ensures that the loss increases exponentially as the predicted value approaches one.

We can see that binary cross entropy error "penalizes confident but incorrect predictions more heavily, which helps the model learn to make more accurate predictions over time"[1]. Mathematically, the error J of the model can be expressed as the mean of losses of every example in the dataset:

$$J = -\frac{1}{n}\sum_{i=1}^{n} y_i \log log\left(\hat{y}_i\right) + \left(1 - y_i\right) log(1 - \hat{y}_i)$$

BUILDING THE MACHINE LEARNING MODEL

A neural network, using interconnected nodes (neurons) similar to the human brain in a layered structure, is a computational model in the machine learning process. The input layer receives the initial data, in which each node represents a feature of the dataset. After the input layer, there are hidden layers that consist of nodes that perform computations on the input data. An output is produced in the hidden layer that passes to the next layer by applying some mathematical functions. The number of hidden layers and the number of nodes in each layer, which are called hyperparameters in neural networks, are set before training the network. The output layer produces the final prediction based on computations performed by the previous layers.

Weights are added in each connection of nodes, and the network adjusts to these weights to minimize the error. Each node applies an activation function to the weighted sum of its inputs. Activation functions, ultimately determining whether to transmit a signal and what to transmit to the next neuron, define the output of the node for a given input or set of inputs.

Activation functions are necessary in most cases because without them, inputting and outputting data in each layer is a process of linear summation, which means the output in the next layer only carries the linear transformation of the last layer inputs. While many problems are much more complex than linear combinations, activation functions allow the neural network to apply to more non-linear models. Therefore, an appropriate activation function improves the outputs of a model and makes the outputs more accurate and desirable.

For this dataset, we consider using Sigmoid and Rectified Linear Unit (ReLU) functions. Sigmoid function maps the weighted sum of inputs to a range of 0 to 1, implying that it's suitable for the binary classification problem; and its gentle slope avoids spread-out outputs, stabilizing the result. The Sigmoid function is shown below.

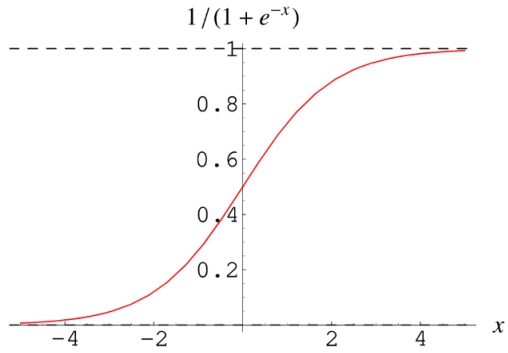

Figure 2. Sigmoid function

A prominent disadvantage of Sigmoid function, however, is the vanishing gradient problem. When the value is very close to 0 or 1, the gradient (rate of changing) is very small. Especially when there are many layers, the model cannot keep being trained because of the vanishing gradient. However, we do not have a very complex dataset so there will not be many layers, Sigmoid function should work well.

The ReLU function is a piecewise linear function, expressed as:

$$f(x) = \{x,\ (x \geq 0)\ 0,\ (x < 0)$$

Hence, it does not have the vanishing gradient problem since its derivative is 1 when the input is positive.

For our dataset, we utilize Sigmoid and ReLU as activation functions and adopt the model with the highest accuracy by changing the parameters (epochs, number of neurons…). Since we have only 14 features, we can start with a small number of neurons. Figure 3 below shows the loss of the model when we have 4 neurons in the ReLU function, and 1 neuron in the Sigmoid function, and the dataset passes through the algorithm 50 times.

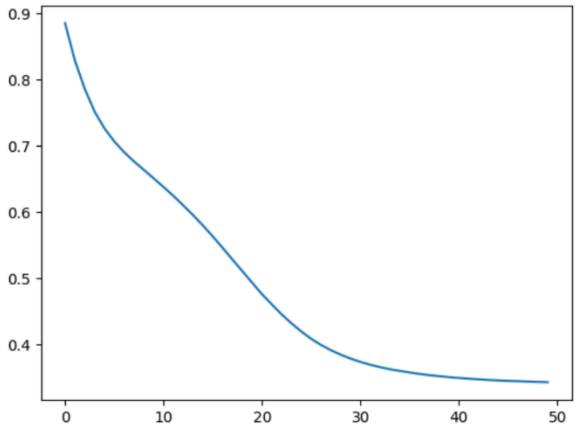

Figure 3. Loss of the model when epochs=50

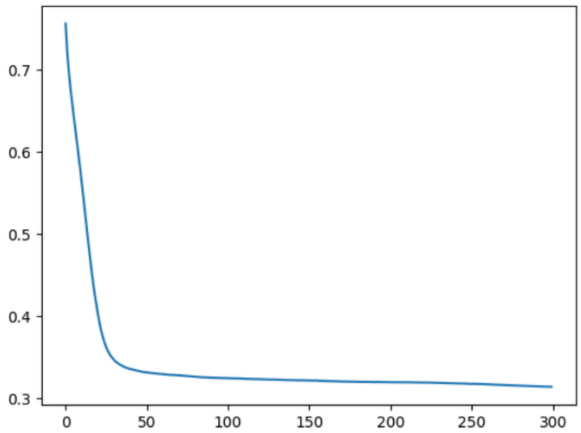

Figure 4. Loss of the model when epochs=200

The incomplete convergence and the instability shown in the diagram imply an insufficient amount of epochs. We increased the epochs value to 200, shown in Figure 4.

The loss is continuously decreasing as the number of epochs increases, the curve tends to converge to a relatively stable or minimal value, and the pattern is smooth. As shown in Figure 5, the accuracies of both training and validation are close to each other and comparatively high. This suggests a desirable model.

	loss	accuracy	val_loss	val_accuracy
0	0.883245	0.525886	0.833443	0.534694
1	0.783366	0.551090	0.747096	0.544898
2	0.710532	0.593324	0.685534	0.593878
3	0.659105	0.632153	0.644437	0.636735
4	0.622275	0.673025	0.615599	0.665306
...	...	...	...	...
195	0.316747	0.856948	0.380048	0.834694
196	0.316629	0.856948	0.379920	0.836735
197	0.316447	0.854905	0.379884	0.836735
198	0.316447	0.857629	0.379753	0.836735
199	0.316350	0.856267	0.380128	0.836735

Figure 5. Accuracies in validation and training sets

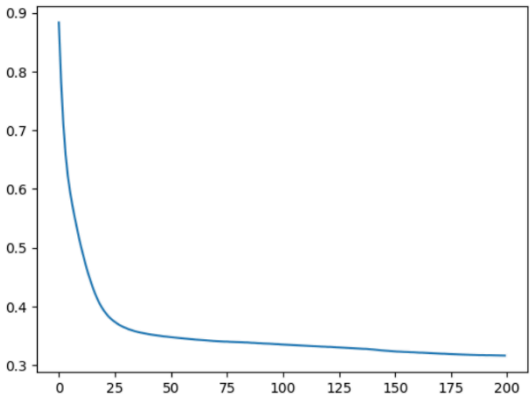

Figure 6. Loss of the model when epochs=300

However, we can continue increasing the number of epochs to see whether the model can perform better. Figure 6 shows how the model performs when there are 300 epochs.

As shown in Figure 7, the accuracy of the validation set increases subtly, while the trained set accuracy stays the same. The validation and trained errors in this case are 0.37 and 0.31, respectively. When we increase the epochs to 400, the error increases, implying a potential overfitting. Hence, we stop at 300 epochs.

	loss	accuracy	val_loss	val_accuracy
0	0.756422	0.517711	0.727914	0.546939
1	0.722103	0.533379	0.701957	0.563265
2	0.697248	0.555858	0.681998	0.579592
3	0.677203	0.572888	0.665433	0.608163
4	0.659858	0.596049	0.650095	0.622449
...	...	...	...	...
295	0.314398	0.856948	0.366251	0.840816
296	0.314282	0.858311	0.366774	0.840816
297	0.314173	0.856948	0.366974	0.840816
298	0.314319	0.855586	0.366925	0.840816
299	0.314124	0.856267	0.367060	0.840816

Figure 7. Loss of the model when epochs=200

MEASURING THE MODEL QUALITY

There are several ways to measure how good the predictions of a model are in classification problems. *Accuracy* is equal to the number of correct predictions divided by the number of total predictions; *recall* is equal to the number of true positives (real positive labels that are also predicted to be positive) divided by the number of true positives plus false positives (real positive labels that are predicted to be false); and *precision* is equal to the number of true positives divided by the number of positive predictions. The *F1-score* is the harmonic mean of precision and recall and provides a balanced measure between the two. It ranges from 0 to 1, where a value of 1 represents a perfect model, and a value of 0 indicates poor performance.

By applying a threshold of 0.5, that is, if a predicted value is greater than 0.5, it will be assigned a value of 1; otherwise, it will be assigned a value of 0; we can get the classification report of the model, shown below.

```
16/16 [==============================] - 0s 2ms/step
Validation error = 0.36705977
46/46 [==============================] - 0s 1ms/step
Training error = 0.31240636
              precision    recall  f1-score   support

           0       0.81      0.95      0.87       739
           1       0.93      0.77      0.84       729

    accuracy                           0.86      1468
   macro avg       0.87      0.86      0.86      1468
weighted avg       0.87      0.86      0.86      1468
```

Figure 8. Classification report of the model

The validation and training errors are low. For class 0, the precision is 0.81, meaning that out of all instances predicted as class 0, 81% were correct. For class 1, the precision is 0.93, indicating that 93% of the examples predicted as class 1 were correct. For class 0, the recall is 0.95, meaning that 95% of the actual class 0 examples were correctly identified. For class 1, the recall is 0.77, indicating that 77% of the actual class 1 examples were correctly identified. The accuracy of the model is the overall percentage of correctly classified instances in the dataset. In this case, the accuracy is 0.86, indicating that the model correctly predicted 86% of the instances in the evaluation dataset. The high F1-scores further indicate a good balance between precision and recall.

Overall, the model appears to perform reasonably well.

APPLICATIONS OF THE MODEL

When applying the model to the auxiliary diagnosis of osteoporosis, we are provided with the patient's information following the features.

For example, we have a new example as shown below, and want to use the model to predict whether this patient has a big possibility of being diagnosed with osteoporosis or not.

```
XNew = [[34, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0]]
```

Figure 9. New set of example

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
[[1]]
```

Figure 10. Output label of the new set of example

The output label is 1, as shown in Figure 10, suggesting that the person is likely to have osteoporosis.

In this case, this patient may consider taking further medical tests, for the doctors to evaluate the state of illness. The model makes it possible for potential patients to apply their data on this model and prevent the further development of osteoporosis at an early stage. Doctors, in turn, can also pay attention to these people diagnosed with 'likely to get osteoporosis', reducing the occurrences of misdiagnosis and missed diagnosis.

CONCLUSION

In conclusion, the model based on supervised learning for the auxiliary diagnosis of osteoporosis is completed. By choosing binary cross entropy error as the loss function, neuron network as the machine learning computational model, and properly training the model, the model presents a high accuracy in prediction, suggesting the prospect of extended application, allowing effective prevention of diseases in advance for patients, improving the diagnosis accuracy for doctors. When new data becomes available after implementing this model, continuous evaluation and iteration of the model are necessary for enhancing data accuracy and applicability. Meanwhile, it is also crucial to explore other more specific features related to osteoporosis and bone health, in pursuing to improve diagnostic accuracy. For example, trabecular bone score, bone microarchitecture parameters, and bone turnover markers; also, noticing that endocrine diseases, connective tissue diseases, chronic kidney diseases, gastrointestinal diseases, and hematological diseases can all cause osteoporosis, a more comprehensive can be achieved through adopting data from other departments as features in the model.

The model is currently only limited to auxiliary and preventive diagnosis. Looking forward, models to evaluate current diagnosis progress and predict further disease development should be considered for patients already diagnosed with osteoporosis, in which medical test results like the bone density in DXA Scan should be analyzed to predict the trend of the disease development. Also, considering the current dataset only includes binary values, the addition of more quantitative data will possibly improve the model's accuracy.

REFERENCE

Agarap, Abien Fred. "Deep Learning using Rectified Linear Units (ReLU)."*arXiv*, arxiv.org/abs/1803.08375.

Godoy, Daniel. "Understanding binary cross-entropy / log loss: a visual explanation." *Medium*, 28 11 2018, towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a.

"Graph logarithmic functions." *Course Sidekick*, www.coursesidekick.com/mathematics/study-guides/ivytech-collegealgebra/graph-logarithmic-functions.

Bailey, Aubrey. "Osteoporosis Facts and Statistics: What You Need to Know." *Verywell Health*, 11 Oct. 2022, www.verywellhealth.com/osteoporosis-facts-5323575#:~:text=Osteoporosis%20most%20commonly%20affects%20Asian%20and%20non-Hispanic%20White,is%20most%20common%20in%20the%20non-Hispanic%20White%20population.