Application of Machine Learning to Detect Bank Note Fraud

By Allen Ting

AUTHOR BIO

Allen Ting is a high school student who developed a keen interest in mathematics since elementary, which has continued to grow. Recently, he has been exploring the field of machine learning and its underlying theories and models. He's particularly intrigued by the practical applications of machine learning, such as its potential to detect banknote fraud. After looking into the theory behind different types of machine learning models and looking into the alarming impacts of counterfeit currency, Allen has worked to identify the key features that ultimately best differentiate a genuine bank note from a fake one. As a young student, he aspires to contribute meaningfully to the field of machine learning and mathematics in the future, aiming to make a practical and widespread impact.

ABSTRACT

We will discuss how to detect whether a bank note is genuine or counterfeit using machine learning. We first will explain the basic concepts and mathematics behind machine learning. Then, using the binary cross-entropy model and logistic regression, we differentiated between real and counterfeit banknotes from a dataset of banknote dimensions.The results showed that the diagonal length was the most important dimension when classifying the bills, with an error of 0.1546 and accuracy of 0.98, and that inputting the length of the bottom margin and diagonal trained the model to have an error rate under 10%.

INTRODUCTION

Counterfeit bank notes are a problem that potentially affects the nation's economy in a significant manner [3,4]. Although counterfeit currencies are typically produced for personal gain, there also exist instances where counterfeiting money was used as a political weapon against rival nations. When a large number of counterfeit bills are in circulation in the economy, inflation rises due and companies lose monetary value to this artificial increase. Genuine bills lose their true value, and as a result, prices will be marked up, forcing everyday people to spend more money on goods and services. Therefore, it is important to create a machine that can identify the legitimacy of a banknote before it is lost in circulation.

Machine learning, also known as artificial intelligence (AI), is a process that constructs models that perform a certain task without being directly programmed to do so [1,2]. The idea is that machine learning algorithms are trained using datasets, which consist of input data, or features, and corresponding output values, or labels. The algorithms learn from these examples by detecting patterns, extracting relevant features, and generalizing from the data to make predictions on new data.

Machine learning is widely used for data analysis and has numerous applications in many fields. In medicine, machine learning is used to predict diseases and illnesses like heart disease and diabetes before it is too late. Machine learning is used to predict stock prices, create self-driving cars, and suggest products and services based on previous purchases. In the banking industry, machine learning can be used to protect private data, decide loan applications, and detect fraud like counterfeit banknotes.

In this paper, we will train a model that can predict whether a swiss banknote is genuine or counterfeit from a dataset containing 100 genuine and 100 counterfeit notes. The dataset came from Kaggle, a website that has numerous datasets that is free for the public [5]. These datasets can be used to train machine learning models.

DATA PREPROCESSING

The dataset consists of 200 Swiss banknotes, 100 that are genuine and 100 that are counterfeit. Although the banknotes in the dataset are real banknotes, the dataset is synthesized and is not extracted from a real sample. An example of the data in the dataset is shown in table 1, which show the dimensions of the first five banknotes and if they are genuine or counterfeit.

Counterfeit	Length	Left	Right	Bottom	Top	Diagonal
0	214.8	131.0	131.1	9.0	9.7	141.0
0	214.6	129.7	129.7	8.1	9.5	141.7
0	214.8	129.7	129.7	8.7	9.6	142.2
0	214.8	129.7	129.7	7.5	10.4	141.0
0	215.0	129.6	129.6	10.4	7.7	141.8
			...			

Table 1: Data for the first five bills in the dataset

The dataset contains 6 features, or characteristics, and 1 label, or output. The 6 features are the following:

x_1 : The length of the bill (Length)
x_2 : The height of the left of the bill (Left)
x_3 : The height of the right of the bill (Right)
x_4 : The height of the bottom margin of the bill (Bottom)
x_5 : The height of the top margin of the bill (Top)
x_6 : The length of the diagonal of the bill (Diagonal)

Figure 1 below shows the measurements for each feature on the Swiss bank note.

Figure 1: Labeled banknote measurements

All measurements are measured in millimeters (mm). The label "counterfeit" states whether the bill with the specified features is genuine (0) or counterfeit (1).

There is no data missing in the 200 banknotes, meaning that it is not necessary to clean the dataset, but we improved the performance of the machine learning algorithm by scaling the data values in the dataset before feeding it into the algorithm. Scaling the data transforms each data value to a number between -1 and 1, thereby reducing the range of the values of the dataset and making it easier for the model to process the information.

TRAINING AND VALIDATION SET

To train and test the model, the data is split into two sets: a training set and a validation set. The training set consists of 75% of the data, while the other 25% of the data is assigned to the validation set. The training set will be used to train and develop the model, and the machine must have both the features (length, left, right, etc.) and the classification (counterfeit or genuine). To measure the accuracy of the model, the validation set will be fed into it without the classification, and the model will then classify whether or not the bill is counterfeit based on the measurements of the features for each bill. It is important to not reveal the classification in the validation set so that we can compute the error for the trained model. If the error is small, then the model will likely make better predictions.

LOGISTIC REGRESSION

One type of machine learning technique for supervised learning problems, the form of machine learning which uses labeled input and output data instead of raw data, is logistic regression. This method only works for binary classification problems. In binary classification problems, the labels, or result, only have two results: 1 or 0. In our dataset, if the result is 1, then the bill is counterfeit. If the result is 0, then the bill is genuine.

Logistic regression is a special case of linear regression because instead of using a line to predict the classification, a logistic function is used instead. This logistic function is also referred to as the activation function. In this paper, the sigmoid function will be the activation function for logistic regression, which is shown in figure 2.

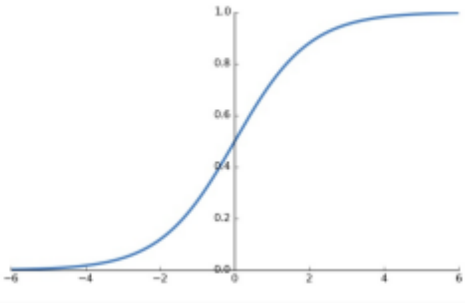

Figure 2: Sigmoid function graph

There are a few properties of the sigmoid function. To begin with, $\sigma(x)$ is an increasing function. Furthermore, $0 < \sigma(x) < 1$ for all x, $\sigma(x)$ becomes arbitrarily close to 0 as x becomes large in absolute value but negative, $\sigma(x)$ becomes arbitrarily close to 1 as x increases, and $\sigma(0) = 0.5$.

In Logistic Regression, the prediction, $\hat{y}$, is of the form

$$\hat{y} = \sigma(w_1x_1 + w_2x_2 + \ldots + w_kx_k + b)$$

where σ is the sigmoid function, k is the number of features in the data set, and $w_1, w_2, \ldots, w_k$, b are parameters that minimize the cross entropy error on the training set. In order to understand how to determine the values of $w_1, w_2, \ldots, w_k$, b, we must understand how to calculate the cross entropy error.

BINARY CROSS ENTROPY ERROR

Binary Cross Entropy is defined as a loss function that measures the difference between the predicted probabilities of a binary classification model and the actual binary labels of the data.

Let $y_1, y_2, \ldots, y_n$ be the labels, out outputs for a set of examples. Let $\hat{y}_1, \hat{y}_2, \ldots \hat{y}_n$ be the model's respective predictions for each example. Because this is a binary classification, the labels y_i must be equal to 0 or 1. However, the predictions $\hat{y}_i$ can be values such that $0 < \hat{y}_i < 1$. The cross entropy error is defined as followed:

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

Note that the closer $\hat{y}$ is to y, the smaller the error. If $\hat{y} = y$, then we can observe that the error is 0. The values of $w_1, w_2, \ldots, w_k$, b are chosen to minimize the values of $\hat{y}$, thereby minimizing the value of J.

These parameters that are selected are those that make mean binary crosentropy error on the training set as small as possible, and this is done through existing and available libraries like Keras. It is not necessary to dive deeper into the algorithms used to find the parameters that make the error as small as possible.

DATASET ANALYSIS

We apply logistic regression to our swiss bank notes dataset to train the model to predict whether or not a banknote is genuine or not given its dimensions.

```
model = 0
model = Sequential()
#model.add(Dense(1,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy')
model.fit(X_train_scaled,y_train,epochs=1000,verbose=0)
J_list = model.history.history['loss']
plt.plot(J_list)
```

Figure 3: Binary Cross Entropy Graph Code

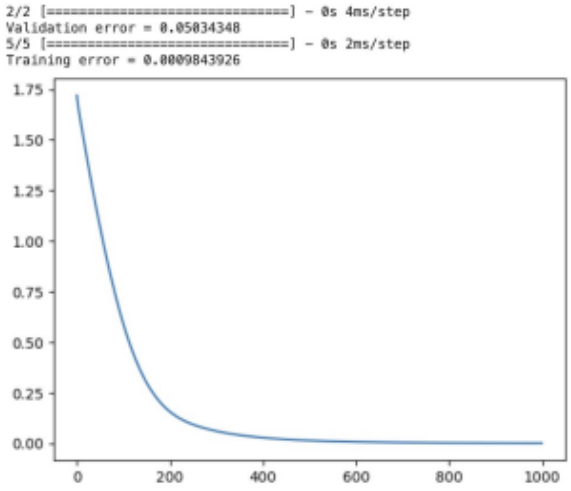

Figure 4: Graph of the error of the model

Figure 3 shows the code used to train the model. In the code, the model reiterates through the scaled training set 1000 times. The number of iterations or cycles the model completes is known as the number of epochs. The model then fits such that it minimizes the error generated from the training set. Figure 4 shows a graph that shows the relationship between the number of epochs and the training error of the model. As the graph flattens out, we stopped cycling through the dataset to prevent overfitting. Overfitting occurs when the model

learns the training data too well, including slight variations and noise in the training dataset. When a model overfits the training dataset, it fails to generalize the new data present in the validation set.

Figure 5: Training and validation error

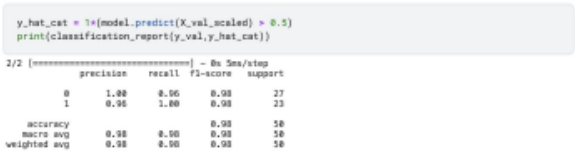

Figure 6: Percent accuracy of the model

The biocrossentropy error when using all the features in the dataset (length, left, right, top, bottom, diagonal) is 0.000253629. On the validation set, the model was able to accurately classify 100% of the genuine bank notes and 96% of the counterfeit bank notes. Therefore, in this given validation set, the model accurately classified 98% of the bank notes in the validation set. The results are shown in figure 5 and figure 6, where figure 5 shows both the training and validation error, and figure 6 shows the accuracy of the model in the chart.

FURTHER INVESTIGATION ON THE FEATURES

To see which features are the most important when training the model, we can isolate each feature and train the model using one feature at a time. Then, by checking the accuracy on the validation set on all 6 features, we can identify which one is the most significant when classifying the bank note.

To do this, we replace the input with one feature in the code and run the code again as is.

Figure 7: Code for one feature

Figure 7 shows the new code used to determine the significance of each feature. By setting the variable X to input one feature at a time instead of all six features, running the same code as the one used on the dataset with all six features produces the accuracy and error for the single feature that was used as the input.

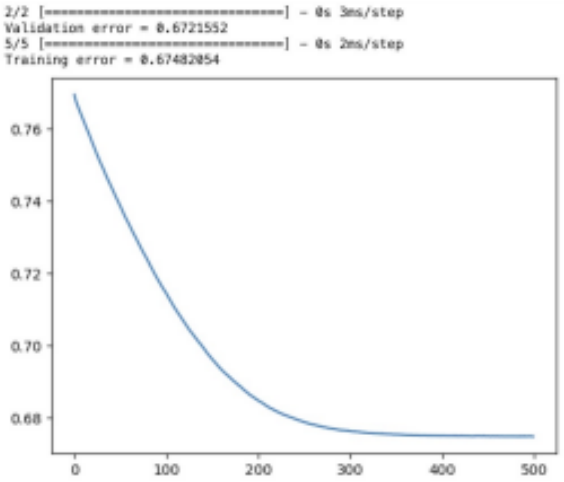

Figure 8: Results after only giving the note's length in mm

Figure 8 shows the error of the model after training it with only the length of the bank notes with an epoch number of 500. After running the code, we get a training error of approximately 0.6748 and a validation error of approximately 0.6725.

This process was repeated for the five other features: Left, Right, Bottom, Top, and Diagonal. Each feature was used to train the model and the training error, validation error, and accuracy for each respective model is shown in table 2.

Feature	Training Error	Validation Error	Accuracy
Length	0.6748028	0.6722918	0.62
Left	0.57226825	0.50991535	0.78
Right	0.49197468	0.46006912	0.82
Bottom	0.27800325	0.3105967	0.92
Top	0.47173423	0.44457167	0.72
Diagonal	0.025608378	0.15463987	0.98

Table 2: Training error, validation error, and accuracy for each feature

These values can be confirmed with a heatmap shown in figure 9 of the correlations between a feature and the label "counterfeit".

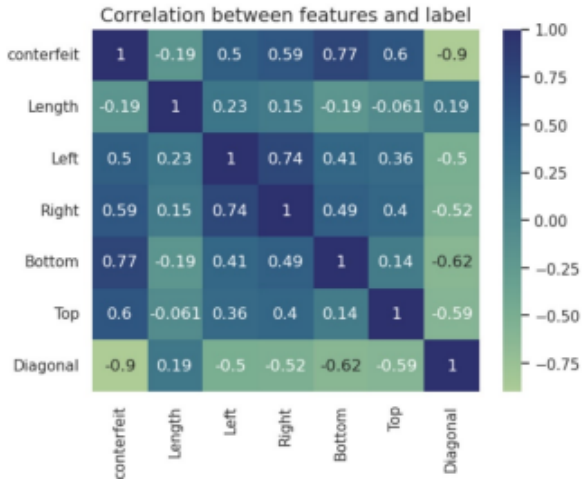

Figure 9: Heat map of the correlation between each feature and the label

The closer the correlation is to -1 or 1, the more related the two variables are. Thus, if the correlation between a feature and "counterfeit" is close to -1 or 1, then the feature is significant when determining whether a bank note is counterfeit or genuine.

Features such as "diagonal" and "bottom" have a high set accuracy (0.98 and 0.92 respectively) and have a low training and validation error compared to the other features. However, their error is high compared to the error we achieved from inputting all the features into the model. We reduced the validation error without inputting all the features by inputting different combinations of pairs of features into the model. To predict which pairs work well together, we use the heatmap and choose the top four features with the greatest absolute correlation value between "counterfeit," pair them up with each other, and feed the two labels into the model to train it.

From Table 3 below, we can observe that the validation error when only diagonal and bottom measurements are inputted are very similar to the validation error when all the features are inputted. Therefore, it is not necessary to use all of the features in the machine learning model. Instead, we only need the dimensions of the bank note's diagonal and bottom to achieve similar results.

Features	Training Error	Validation Error	Accuracy
Diagonal, Bottom	0.008380662	0.062047623	0.98
Diagonal, Top	0.0068245945	0.14657234	0.98
Diagonal, Right	0.009925004	0.13944182	0.98
Bottom, Top	0.04957412	0.088041	0.96
Bottom, Right	0.23715912	0.24233983	0.92
Right, Top	0.36362162	0.32460323	0.82
Right, Left	0.49034384	0.45351523	0.82

Table 3: Training error, validation error, and accuracy for the given pairs of features

APPLICATION OF ML ON BANK NOTE FRAUD

The results show that the logistic regression algorithm is able to accurately predict and classify whether or not a bank note is counterfeit by looking for recurring trends in the dataset. In the end, the model accurately classified 98% of the banknotes correctly with two features, the diagonal dimension and bottom margin dimension with a validation error of 0.062. The high accuracy and low error shows that the model has the potential to classify other sets of bank notes with high accuracy and low error.

Since the model also only needed two features instead of all six, the model may not

need many dimensions of the banknotes in future datasets if there is a high correlation between one specific dimension and the classification. Requiring fewer features makes data collection less time consuming and expensive.

REFERENCE

Brown, S. (n.d.). Machine learning, explained. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained (accessed: 06.21.2023).

Francis, R. F. A. (n.d.). A brief history of currency counterfeiting. https://www.rba.gov.au/publications/bulletin/2019/sep/a-brief-history-of-currency-counterfeiting.html (accessed: 06.23.2023).

IBM. (n.d.). What is machine learning? https://www.ibm.com/topics/machine learning (accessed: 06.29.2023).

INTERPOL. (n.d.). Counterfeit currency [fact sheet]. https://www.interpol.int/content/download/10475/file/Counterfeit$%5C%$20currency.pdf (accessed: 06.19.2023).

Riedwyl, B. F. H. (n.d.). Multivariate statistics: A practical approach. london: Chapman hall, tables 1.1 and 1.2, pp. 5-8. https://www.kaggle.com/datasets/chrizzles/swiss-banknote-conterfeit-detection (accessed: 06.10.2023).