Dimensionality-Reducing Encoding for Classification of Pythagorean Engendered Numbers

Dead Duck or Phoenix?

π, 2019

Rany Tith
Spark Tank
Riverside, CA, 92507
rany.tith@protonmail.com

Oscar I. Hernandez
ARKS
Riverside, CA, 92507
ohernandez13@simons-rock.edu

Abstract

We apply modern well-known machine learning classifiers to the problem of determining whether a given positive integer is even or odd, a problem dating as far back as 6th century Classical Greece before common era and even further in Ancient Egypt. We prove that the classification of engendered numbers is possible. This is done by utilizing a unique dimensionality-reducing encoding that was implemented before the machine learning models were trained. Overall, our models' results proved to be successful in classification as indicated through AUC and ROC analysis.

1. Introduction

In this article, we'll discuss a partition of numbers discovered by the Ionian Greek philosopher Pythagoras (c.570–c.495 BC), which he discovered during his tenure in Egypt before founding a school of math in Greece. "To him, the odd numbers were male, and the evens were female"[PYT].

Following Pythagoras, we will restrict our attention from the usual integers, $\mathbb{Z} = \{..., -1, 0, 1, ...\}$, to only the positive integers, $\mathbb{Z}^+ = \{1, 2, 3, ...\} \subset \mathbb{Z}$. Although one can evaluate whether a small number is even or odd, the general problem remains open.[1] In the following sections, we will formally define even and odd numbers, build a classifier, evaluate its results in a case study, discuss potential improvements, and conclude with relevant open problems.

[1] The authors have attempted to communicate with him, but have been informed that he is not an advocate or user of computers or electronic mail.

2. Preliminaries

2.1 Even and Odd

Definion 1. *[MNT] Let $n \in \mathbb{Z}^+$ be a positive integer. We say that n is even* iff[2] *$\exists k \in \mathbb{Z}$ such that $2 \cdot k = n$ Similarly, n is odd* iff *$\exists k \in \mathbb{Z}$ such that $(2 \cdot k) + 1 = n$.*

For example, $1 = 2 \cdot 0 + 1, 3 = 2 \cdot 1 + 1, 5 = 2 \cdot 2 + 1, 9 = 2 \cdot 4 + 1, 7 = 2 \cdot 3 + 1$ are odd and $2 = 2 \cdot 1, 4 = 2 \cdot, 6 = 2 \cdot 3, 8 = 2 \cdot 4, 10 = 2 \cdot 5$ are even. We have manually classified the next 60 numbers. In the next section, we will use that labeled data to train a classifier to evaluate the even-ness/odd-ness of a given number.

2.2 Receiver Operating Characteristics [ROC]

This section is pulled directly from the source cited. A receiver operating characteristics (ROC) graph is a technique for visualizing, organizing, and selecting classifiers based on their performance.

Definion 2. *ROC graphs are two-dimensional graphs in which tp (true positive) rate is plotted on the Y axis and fp (false positive) is plotted on the X axis.*

They depict relative tradeoffs between benefits (tp) and costs (fp). We'll see the tradeoffs in Section 4.

3. Solution

We propose to use real data such as seen in Figure 5 to train machine learning classifies using the Python library `scikit-learn`. In particular, we will implement support vector machines (SVM), multi-layer perceptrons (MLP), decision tree classifiers (DTC), as shown in Figure 1.

The code is freely available at the first author's GitHub repository. [DRECPEN]. All the results achieved in section 4 were performed on a x86-64 Arch Linux Thinkpad running Python 3.7.2.

3.1 Encoding

In particular, we pre-process the information by encoding the positive integer in binary [AB]. We optimized this encoding and reduced its size to a single bit by restricting our data to the rightmost bit and disregarding the rest of the binary string, since memory becomes a concern when dealing with large numbers. The authors are not in agreement as to why this yields such accurate results, but they are proud of its size and speed.

[2] if and only if

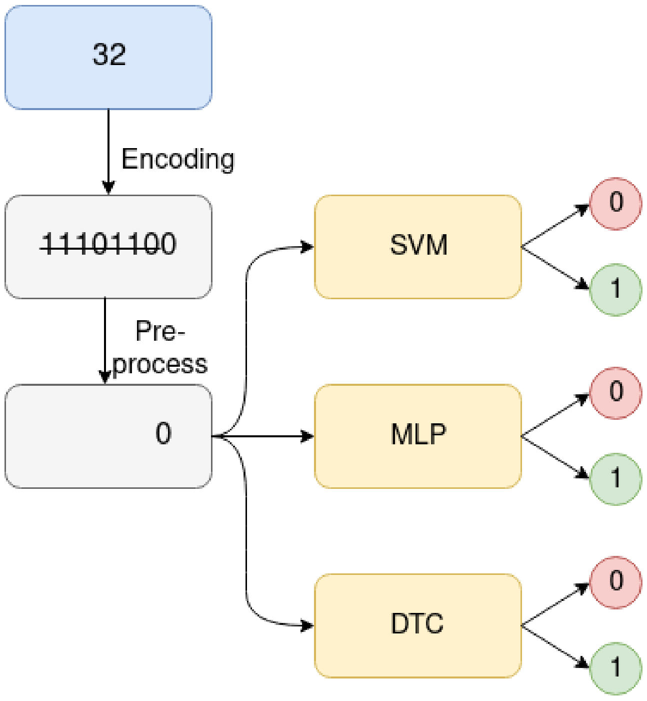

Figure 1. Flow diagram of classification process

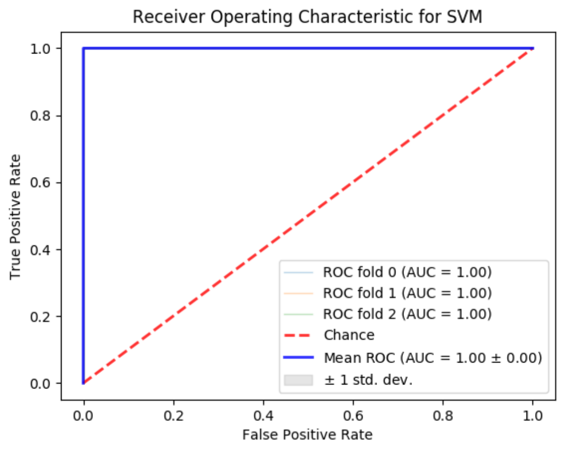

Figure 2. SVM ROC analysis

4. Case Study: Classification Models

4.1 SVM

4.1.1 Discussion

SVM is a popular machine learning model created by Vapnik and Chervonenkis in 1963 which has been implemented successfully in areas such as image classification and hand written character recognizition. The SVM method employs to minimize the equation[SVM]:

$$[1/n \sum_{i=1}^{n} max(0, 1 - y_i(w \cdot x_i - b))] + \lambda |w|^2$$

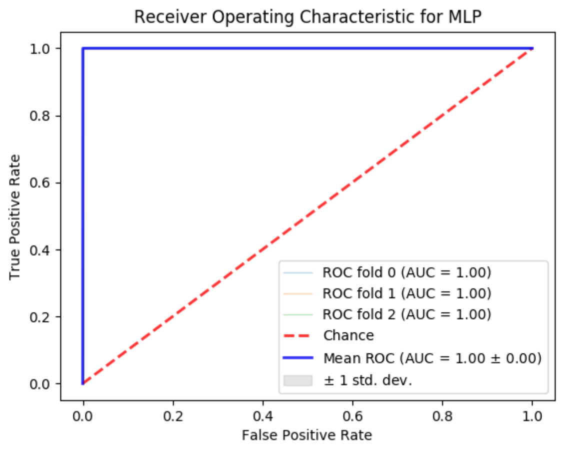

Figure 3. MLP ROC analysis

where w denotes the weight, x is the input variable, y is the dependent variable and λ indicates the margin strength for classification.

4.1.2 Evaluation

In our evaluation as seen in Figure 2 ROC analysis indicates a strong informative model. This is shown as the mean ROC is above the chance line with a low standard deviation. Furthermore, the AUC indicates 1.00 leaving us to conclude that it does better than random chance.

4.2 MLP

4.2.1 Discussion

MLP models are considered to be a type of deep learning that has found to be useful in fields such as speech recognition and image recognition. The model employs layers of neuron that contains the following ReLU activation function:

$$f(x) = max(0, x)$$

Each layer in the neural network contains a number of nodes where a weight w_{ij} was applied at each level to each node.

Learning was then done using back propagation such that:

$$e_j(n) = d_j(n) - y_j(n)\epsilon(n) = 1/2 \sum_j e_j^2(n)$$

d is the target value, y is the output of the perceptron.

And gradient descent was then used to optimize the weights:

$$\Delta w_{ji}(n) = -\nu \frac{\delta \epsilon(n)}{\delta v_j(n)} y_i(n)$$

Where ν is the learning rate, v_j is the sum of all the nodes input, y_i is the output of the previous neuron[MLP], n is the data point, j is the position of the outpude node, and e is the error.

4.2.2 Evaluation

In our evaluation as seen in Figure 3 ROC analysis indicates a strong informative model. This is shown as the mean ROC is above the chance line with a low standard deviation. Furthermore, the AUC indicates 1.00 leaving us to conclude that it does better than random chance.

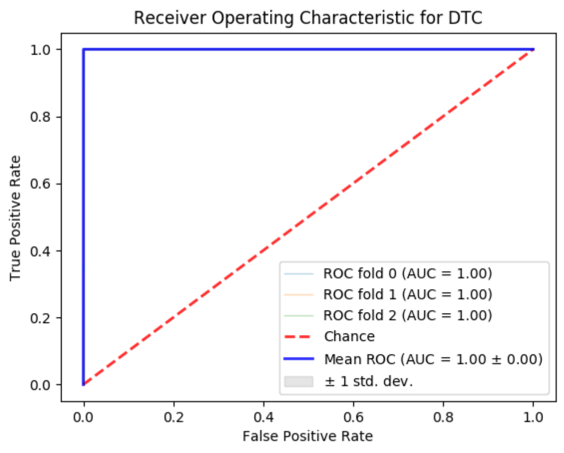

Figure 4. DTC ROC analysis

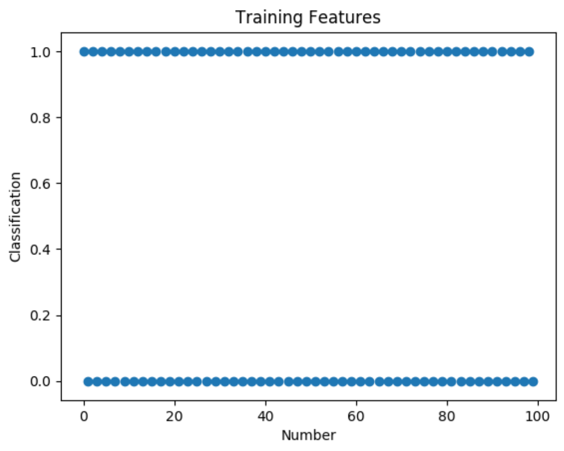

Figure 5. Example training data

4.3 DTC

4.3.1 Discussion

DTC is a popular white model method in use cases such as fraud detection, direct marketing, and economics. More generally we used a decision tree learning method with the information gain metric of:

$$H(T) = I_E(p_1, p_2, ..., p_j) = -\sum_{i=1}^{J} p_i log_2 p_i$$

where p_i are the fractionals that add up to the class that is present in each node that happens at each split in the tree[DTC].

4.3.2 Evaluation

In our evaluation as seen in Figure 4 ROC analysis indicates a strong informative model. This is shown as the mean ROC is above the chance line with a low standard deviation. Furthermore, the AUC indicates 1.00 leaving us to conclude that it does better than random chance.

5. Discussion

The results are truly surprising. They prove that it is possible to classify numbers as being even or odd, a skill that even intelligent machines such as humans do not acquire naturally nor easily. Perhaps this issue is deeply tied to the fact that numbers, as Pythagoreas believed, were engendered, and humans often struggle with evaluating gender while neural networks (e.g. convolutional) find great success.

6. Conclusion

Although this problem is difficult (perhaps intractable), similar problems may be solvable. It may be worthwhile to classify prime numbers, Mersenne primes, perfect numbers, positive numbers, negative numbers, zero, and powers of two.

6.1 Acknowledgements

We would like to thank our colleague, a student at the Washington Elementary School for classifying the first 70 numbers as even or odd. This work was partially supported by Spark Tank, who provided us with coworking space and toiletries. Most importantly, we would like to thank our families, without which we could have completed this work 2 years sooner.

References

[ROC] T. Fawcett An Introduction to ROC Analysis *Elsevier Pattern Recognition Letters*. 2006. https://people.inf.elte.hu/kiss/11dwhdm/roc.pdf

[MNT] K. Ireland & M. Rosen A Classical Introduction to Modern Number Theory *Springer Graduate Texts in Mathematics*. 1990. https://www.springer.com/us/book/9780387973296

[DRECPEN] R. Tith & O. Hernandez Accompanying Source Code *GitHub*. 2019. github.com

[AB] G. Leibniz Explication de l'Arithmtique Binaire *Die Mathematische Schriften*. 1703. http://www.leibniz-translations.com/binary.htm

[SVM] C. Cortes & V. Vapnik Support-vector networks *Springer Machine Learning*. 1995. https://link.springer.com/article/10.1007%2FBF00994018

[DTC] L. Breiman et al. Classification and regression trees *Wadsworth & BrookeCole Advanced Books & Software*. 1984. https://www.crcpress.com/Classification-and-Regression-Trees/Breiman-Friedman-Stone-Olshen/p/book/9780412048418

[MLP] F. Rosenblatt Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms *Springer Brain Theory* 1961. https://link.springer.com/chapter/10.1007/978-3-642-70911-1_20

[PYT] W. Burkert Lore and Science in Ancient Pythagoreanism *Harvard University Press*. 1972. http://www.hup.harvard.edu/catalog.php?isbn=978067453918