Machine Learning Methods applied to the Collatz Conjecture

DSC 680 Summer 2021

Chris Briggs

Abstract

We explore the Collatz Conjecture, a notorious unsolved problem in pure mathematics, using the tools and techniques of data science. We use the case of the Collatz Conjecture as a means of studying various models specifically and data science practice generally.

Background

The Collatz conjecture is a 75-year-old unsolved problem in mathematics, specifically in number theory ("Collatz conjecture," 2021). The famous number theorist and combinatorist Paul Erdös, well known for discovering and tackling hard problems, once remarked about the Collatz conjecture, "Mathematics may not yet be ready for such problems" (Bellos, 2017) ("Paul Erdos," 2021).

The conjecture is as follows: if you begin with a positive integer, then halve it if it is even or triple and add one if it is odd, and continue in this way iteratively, then eventually you will arrive at the number 1.

The problem predates modern computing. Being a problem in pure mathematics, the bulk of attempts on it have been by human and hand. However, there is some precedent in the application of computing techniques to solve problems in pure mathematics. A famous and early example is the original proof of the four coloring theorem thing here (Appel & Haken, 1989).

A breakthrough on a very old unsolved problem was recently made using very simple statistical computing techniques (Oliver & Soundararajan, 2016). The authors noted that, apart from the numbers 2 and 5, all primes are odd and end in either the digit 1, 3, 7, or 9. They computed the likelihood of a prime to end in each digit given the last digit of the previous prime, and to everyone's surprise, found the probabilities to be nonuniform. These results were widely reported by the media (Yirka, 2016).This shows that there is yet low-hanging fruit in terms of leveraging modern computing techniques and power to gain new perspectives on old problems.

Some statistical tests have been brought to bear on the Collatz Conjecture (Granville, 2020), but to the author's knowledge at time of writing, no attempt has been made to model the sequence of Collatz lengths using machine learning techniques, as is done in the present study.

Statement of the Problem

Can machine learning models be used to predict the Collatz length of integers? Which models are successful, and why?

Analysis Approach

First, the data must be generated. A function was written which accepts an integer, and returns its Collatz length. This is done by determining the parity of the integer, then applying the appropriate operation, and iterating a counter per operation. When the integer 1 is reached, the function returns the counter.

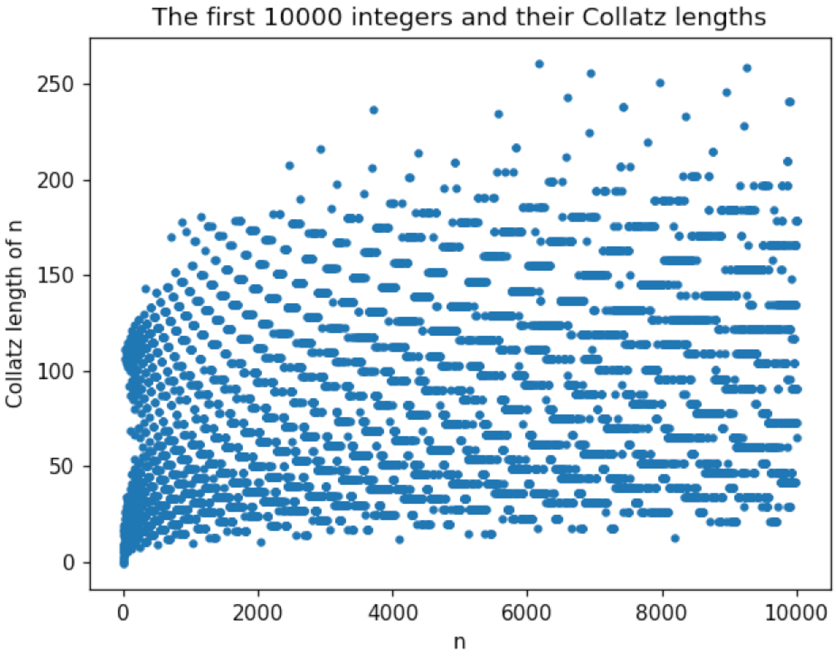
The data was randomly split into training and testing sets, as is standard practice. This detail will be important in the conclusions.

After some exploratory analysis of the data, models were fit to the data. The accuracy was computed on both the training and the testing sets.

Next, the relative accuracies of the models were analyzed, and reasons sought for the discrepancies found. This investigation involved computational metrics, such as percentage accuracy of predictions, as well as graphical investigation to assist the generation of insight.

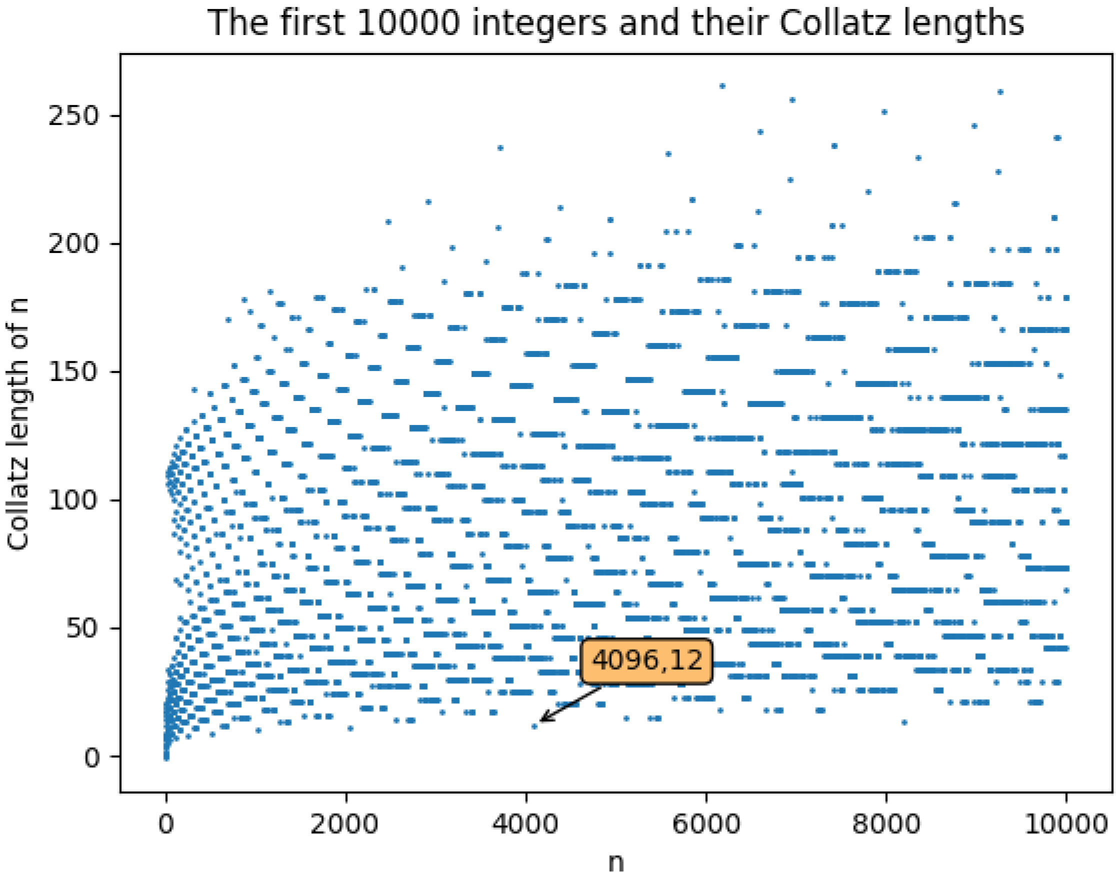
Exploratory Analysis

The data have only one feature. Once generated, the data can be graphed for convenience of perception:



Tantalizingly, there are apparent curves sweeping downward from the upper left to lower right, and also upward from the lower left to the upper right. This is fascinating for the reason that, if the rule by which numbers belong to a given curve could be discovered, then all numbers might be found to belong to such a curve. That would effectively solve the conjecture.

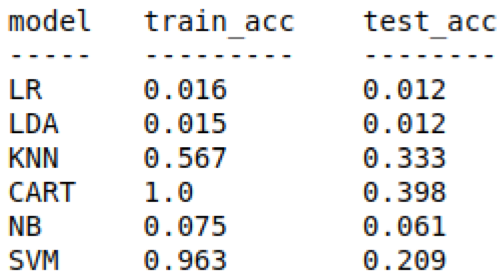
Of course, the problem is understood to be nearly impossible, so this is an unlikely approach to work, and I did not spend much time on it. But it is a direction for further analysis to extend the project. Towards understanding the curves, I created an interactive graph which enables the user to hover over points and read their coordinates. Of course, the interactive functionality will not translate to a PDF, but here is a screenshot:



By exploring these curves on the interactive plot, it can be seen that the upward sweeping curves increase in height by 3 each time, and the downward sweeping curves decrease by 3 each time.

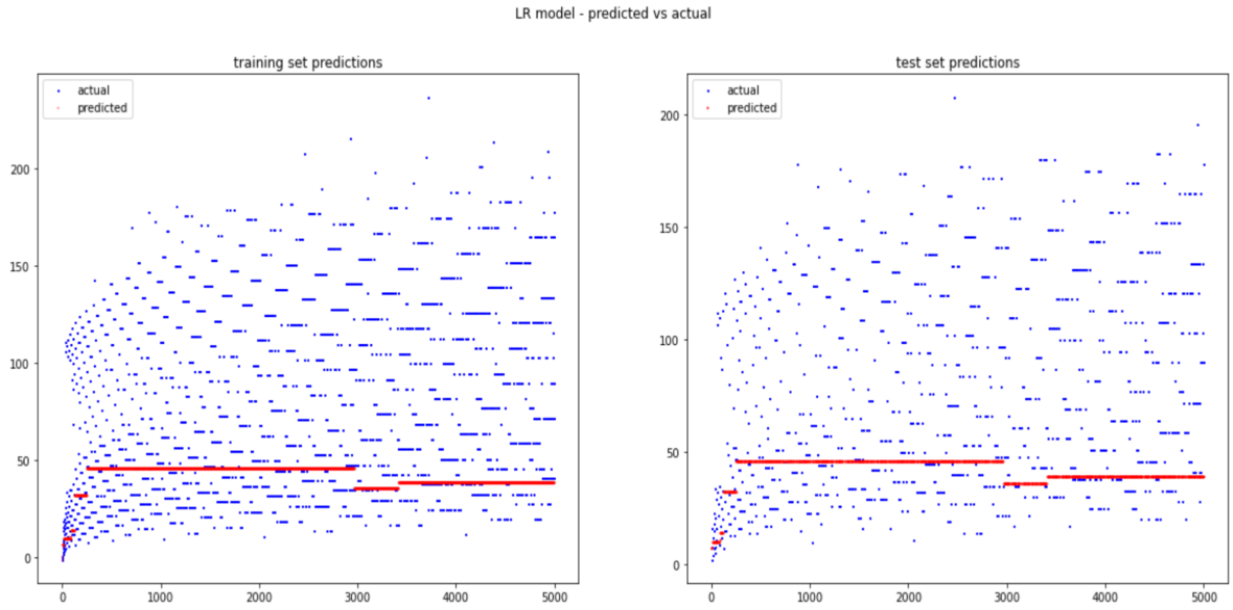
Analysis Results

Six model types were fit to the data: logistic regression, linear discriminant analysis, K-neighbors classifier, decision tree classifier, Gaussian naive-Bayes, and support vector machine. Their accuracies on the training and test data are as follows:

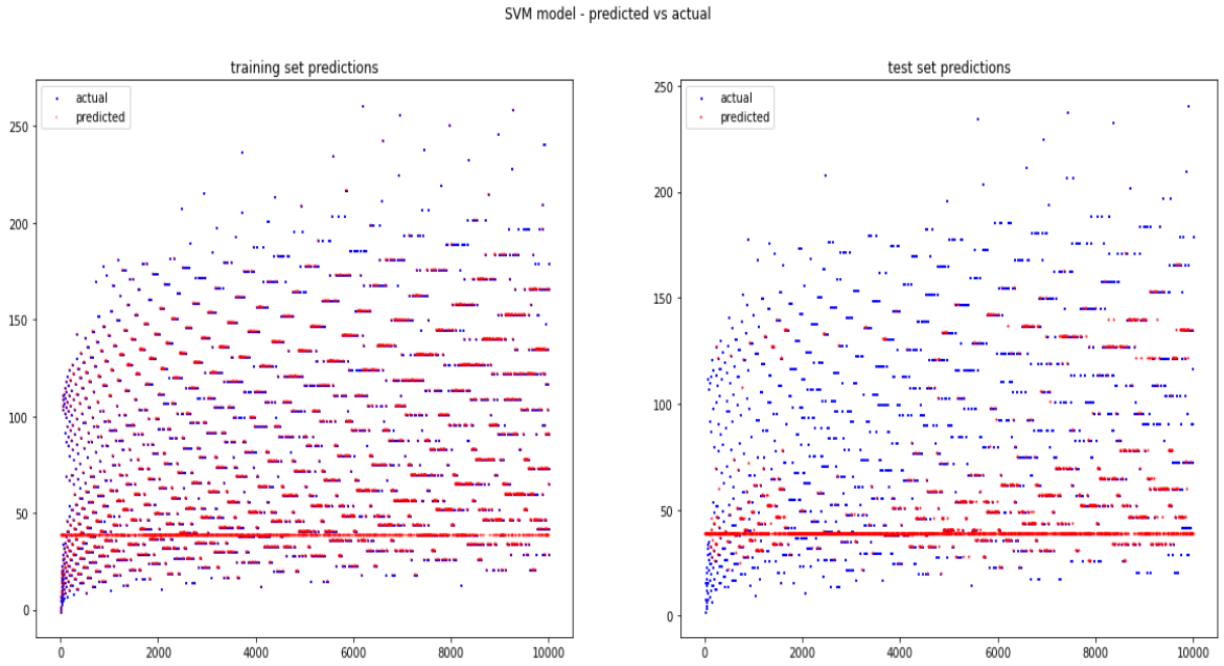


```
model   train_acc    test_acc
-----   ---------    --------
LR      0.016        0.012
LDA     0.015        0.012
KNN     0.567        0.333
CART    1.0          0.398
NB      0.075        0.061
SVM     0.963        0.209
```

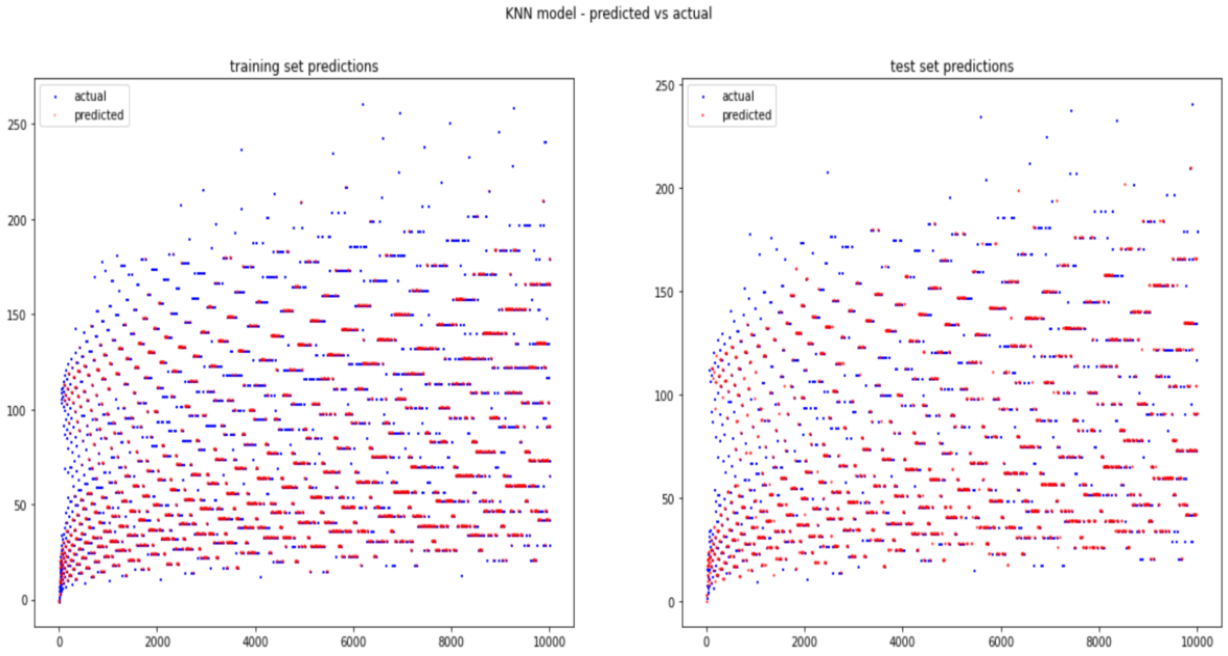
KNN and CART (decision tree) did very well! The others, less so. We turn to visuals of the data overlaid with the various models' predictions in order to gain further insight.



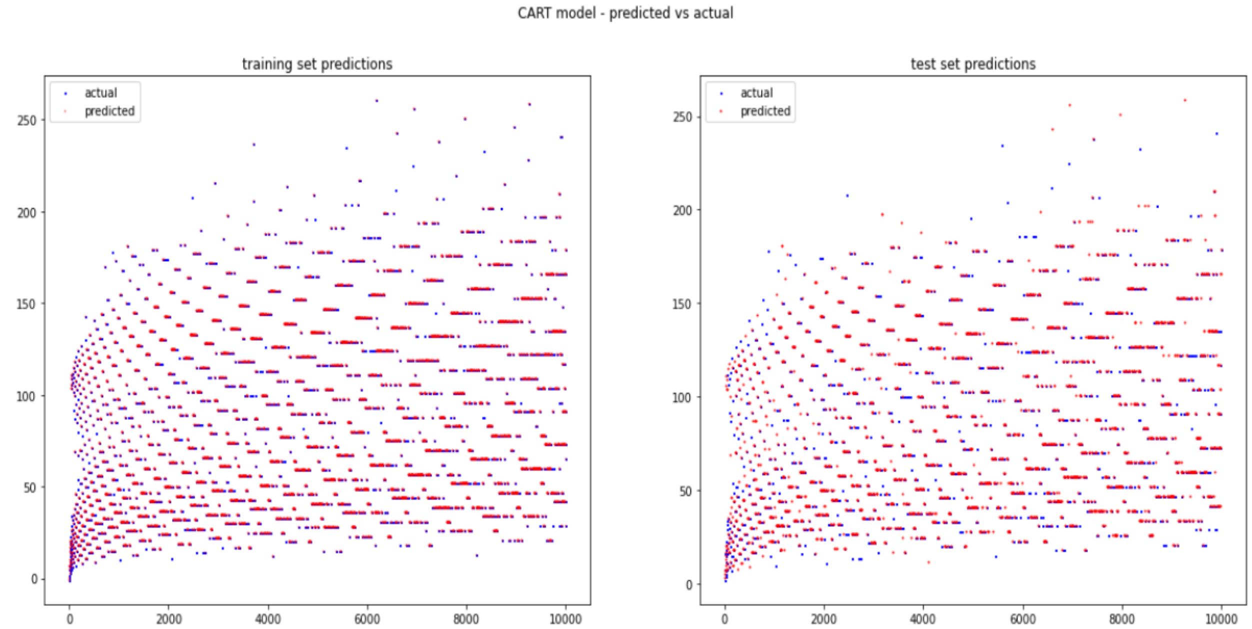
The Logistic Regression model did a pretty poor job on both the training and test set.



The support vector machine model fit well to the data overall, but this was mostly due to overfitting on the training data. When restricted to the test data only, the predictions are much less accurate.



The KNN algorithm did much better with fitting on the testing set. This is exciting, and suggests that KNN may have a bead on the Collatz problem! But more investigation is warranted.



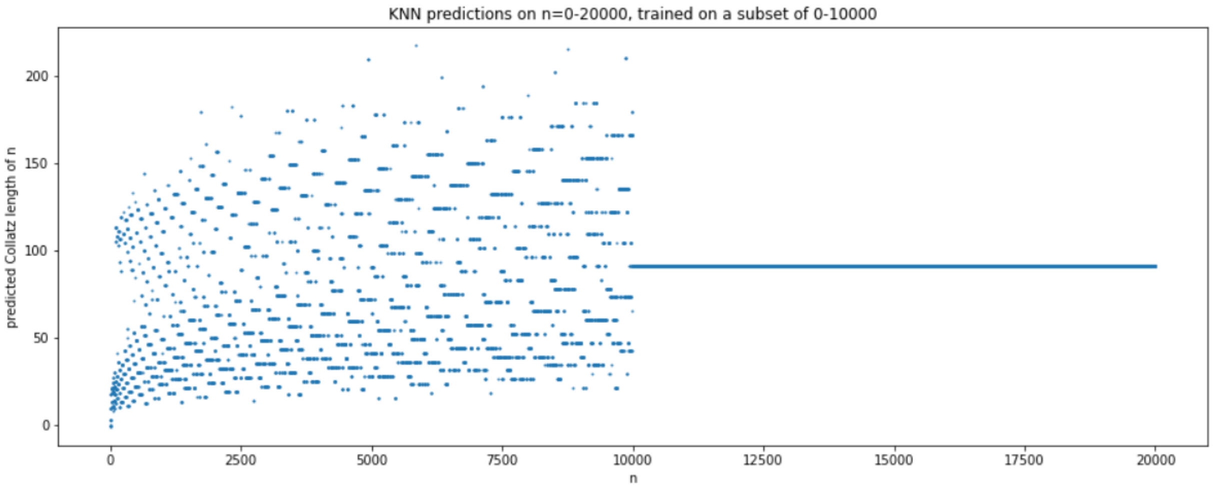
Likewise, the CART model is showing promising results on the test data.

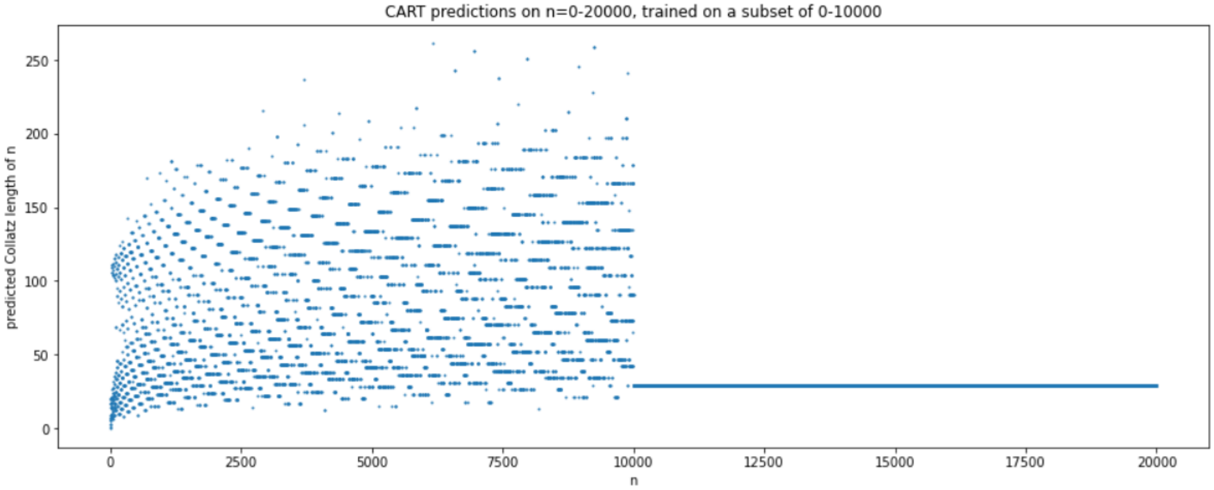
If we understand the basics of the KNN algorithm ("K-nearest neighbors algorithm," 2021), then we can see what KNN is doing by observing that it's good at hitting the long bars but it misses single dots. We might also infer that CART, being a decision tree, is likely making its decisions in a similar way to KNN: using rules about nearby points to say that a point in the test set should have the same length as its neighbors ("Decision tree learning," 2021).

Interestingly, CART is also capable of picking up some individual points when we view the graph of the full data set. See, for example, when the input integer is a power of two (hence has Collatz length lower than its immediate neighbors). However, when we view the graph of the CART predictions on just the test set, it appears to fail to pick up these individual points.

The other models are less accurate in general, and we can see the poor matching between their predictions and the actual lengths in these plots.

If our explanation of the success of KNN and CART is correct, then we should expect to see poor generalizations of these models to numbers larger than those in the training set.





Indeed, we see that both the CART and KNN models are totally incapable of making reasonable predictions above the numbers in the training data set.

Conclusion

Unfortunately, the Collatz conjecture has resisted the predictive power of the suite of models tried against it. But we learned something about the models in the process.

Machine learning models are not interchangeable. The differences go beyond the linear and nonlinear. It is possible to find data sets on which some models perform very well, while others perform poorly. It would be an interesting exercise to take this to the extreme, and find the theoretical maximum differences in performance of different models on a given data set.

This study underscores a need when analyzing a data set which is often overlooked: it is sometimes insufficient to break a data set into training and testing chunks, then train on the one and test on the other. Sometimes it is necessary to acquire new data and test the model against the new data. This is especially true, for example, with time series data. The data analyzed here is not a time series, but analogies can be drawn between this data and time series.

Challenges Encountered

By nature, machine learning algorithms perform computations that are beyond the scale in terms of complexity of what a human can perform by hand. For this reason, a precise understanding, if this means knowing every detail and nuance, of an algorithm will mostly remain elusive. It was therefore challenging in this project to understand the exact function of the algorithms. This challenge was met by confronting performance differences on the training and testing sets with heuristic knowledge of the general ideas behind the algorithms, and verified by extending beyond the original training set.

Limitations

The modeling conducted here has, unfortunately, nothing to say about the truth or falsity of the Collatz conjecture. Although the patterns observed in the data during EDA are tantalizing, not too much should be concluded from their apparent existence. Indeed, it may be true that nearly all numbers fall on such curves, but a single counterexample may still exist which does not. In that way, perfectly modelling the curves may still leave us little closer to determining the truth of the conjecture.

Moving Beyond (how it can be "taken to the next level")

It remains to attempt accurate predictions beyond the highest integer used in the training set. Perhaps modelling of the upward- and downward-sloping curves observed in the data set would be successful here.

As mentioned before, the decision tree model is probably good here because of an elaborate sequence of if/thens that narrow down on the horizontal strips present in the original EDA graphs. I have observed that the horizontal clusters of points become elongated further to the right in the data set. One way to test this hypothesis about the decision tree performance would be to train and test on a set of numbers in, say, the range [1000000:1010000]. If the hypothesis is correct, the decision tree should gain in accuracy, and this should be true even if the model is held to a fixed size (limited in its number of branches).

References

1. Alessandretti, L, Baronchelli, A., He, Y.H. Machine Learning meets Number Theory: The Data Science of Birch-Swinnerton-Dyer. arXiv:1911.02008
2. Appel, K. and Haken, W. (1989) Every Planar Map is Four Colorable. Contemporary Mathematics (98) DOI: http://dx.doi.org/10.1090/conm/098
3. Bellos, A. (2017, July 10). Mathematics has a bad hair day. Retrieved from https://www.newscientist.com/article/2139238-mathematics-has-a-bad-hair-day/
4. Carnielli, W. (2015) Some Natural Generalizations Of The Collatz Problem. Applied Mathematics E-Notes (15).
5. Collatz conjecture (n.d.). Retrieved July 16, 2021 from https://en.wikipedia.org/wiki/Collatz_conjecture

6. Decision tree learning (n.d.). Retrieved July 18, 2021 from https://en.wikipedia.org/wiki/Decision_tree_learning
7. Granville, V. (2020, Feb 29). State-of-the-Art Statistical Science to Tackle Famous Number Theory Conjectures. Retrieved from https://www.datasciencecentral.com/profiles/blogs/state-of-the-art-statistical-science-to-address-famous-number-the
8. K-nearest neighbors algorithm (n.d.). Retrieved July 18, 2021 from https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
9. Lemke Oliver, R.J., Soundararajan, K. (2016) Unexpected biases in the distribution of consecutive primes. https://arxiv.org/abs/1603.03720
10. Paul Erdos (n.d.). Retrieved July 16, 2021 from https://en.wikipedia.org/wiki/Paul_Erdos
11. Yirka, B. (2016, Mar 15). Mathematician pair find prime numbers aren't as random as thought. Retrieved from https://phys.org/news/2016-03-mathematician-pair-prime-random-thought.html

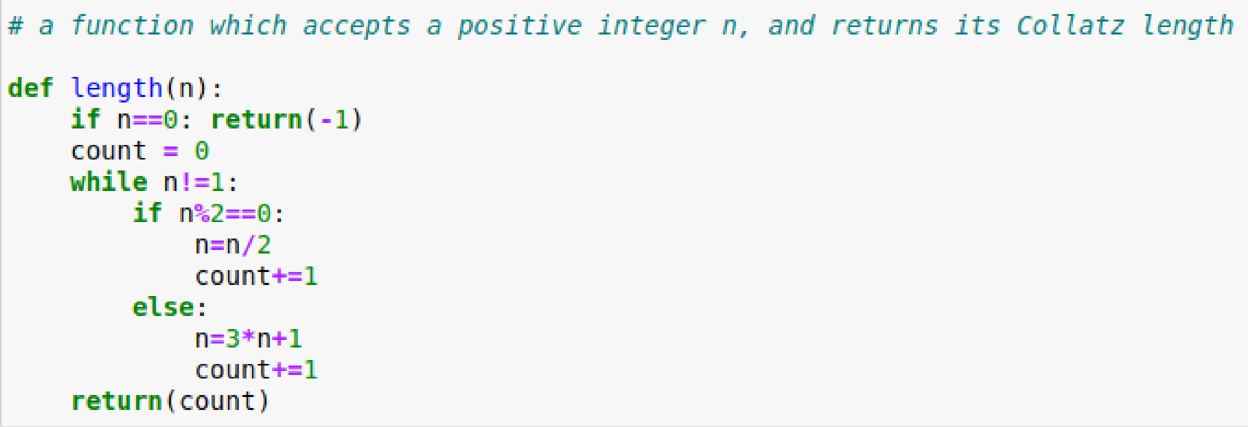
Appendix A.

The sequence of numbers when beginning with 7 is

7 is odd, so we do 3*7+1=22
22 is even, so we do 22/2=11
11 is odd, so we do 3*11+1=34
34 is even, so we do 34/2=17
17 is odd, so we do 3*17+1=52
52 is even, so we do 52/2=26
26 is even, so we do 26/2=13
13 is odd, so we do 3*13+1=40
40 is even, so we do 40/2=20
20 is even, so we do 20/2=10
10 is even, so we do 10/2=5
5 is odd, so we do 3*5+1=16
16 is even, so we do 16/2=8
8 is even, so we do 8/2=4
4 is even, so we do 4/2=2
2 is even, so we do 2/2=1, and we have arrived at 1

Appendix B.

The function used to generate the data:



```python
# a function which accepts a positive integer n, and returns its Collatz length

def length(n):
    if n==0: return(-1)
    count = 0
    while n!=1:
        if n%2==0:
            n=n/2
            count+=1
        else:
            n=3*n+1
            count+=1
    return(count)
```

Ten Anticipated Questions

1. **The decision tree model was most accurate on the data you generated. Is there any indication it would be most accurate on a larger dataset?**
   Perhaps it would be. But we saw in our graphs that none of the models perform particularly well when we extend the data set. We can interpret the fact that the decision tree model performed best on this data set by thinking about how it generates predictions, and this helps us see why the accuracy drops precipitously on extended data sets.

2. **Would it be beneficial to eliminate a subset of the numbers that don't "fit well" to the models, perhaps to understand those curves in the scatterplot better?**
   It's true the stray points probably throw off the modeling of the curves. However, getting the curves perfectly modeled isn't the ultimate goal - the ultimate goal would be to conclude that *each point lies on at least one curve*. So, discarding points which do not appear (in our limited view) to belong to a curve would not assist in solving the conjecture.

3. **Does the data give any evidence for the truth or falsity of the conjecture?**
   Again, not really. The distribution of the points suggests that there are underlying groupings of the integers happening here. That is very tantalizing, and if we were able to understand these groupings ("curves"), we might be able to prove that every integer belongs to at least one. But "most integers belong to one of these curves" is not strong evidence for the truth of the conjecture. Of course, we have not uncovered evidence for its falsity either. I cannot imagine what evidence would do this apart from discovery of a counterexample.

4. **What would it take to disprove the conjecture?**
   Disproof of the conjecture could be very simple. All that is required is a counterexample. That would be a number which either grows indefinitely under applications of the Collatz function. or else gets into an infinite loop (which does not involve the number 1).

5. **How far have people searched for such a counterexample? Why didn't you just make a script to try numbers until you found a counterexample, if there is one?**
   Computers have been used to check all numbers up to 2^68, which is about 295,000,000,000,000,000,000.

6. **You mentioned modelling the curves as a potential extension of the project. Do you have any idea as to where to start?**
   Logarithms. At least for the upward sweeping curves, the very bottom of this curve is made up of the powers of 2 (1, 2, 4, 8, 16, and so on). The Collatz length of 2 is 1, of 4 is 2, and more generally, of 2^n is n. That means this lowest curve is exactly a logarithmic

function. So I would begin with logarithmic functions to try to fit to the other upward curves, and I would also try them for the downward curves.

7. **Why times 3 plus 1? Why not times 5 plus 1?**
   Indeed, these are the numbers Collatz used. There are some generalizations of the conjecture which have been studied. With different choices of constants, some of these problems do indeed have cycles (what would be a negative answer for the Collatz conjecture) (Carnielli, 2015).

8. **You mentioned the need to get new data to test on before being sure that a model generalizes. I've heard that about time series, but time is no element here. Can you elaborate on why this applies here?**
   This comes down to the principle: your sampled data set should not be special, meaning should be totally randomly selected, from the larger set to which you plan to apply the results. This is a classic problem for time series data because you generally want to predict future data points, but you have no access to future data for modelling, so you are necessarily not taking a fair sample of the data. It can't be helped. A similar thing happens when you train on a set of numbers - any set - and try to generalize to all numbers. The set you trained on was finite, and it represents exactly 0% of all numbers.

9. **Is there a way to modify the decision tree or KNN models to better generalize to larger numbers?**
   Probably not. These models look surprising and very tantalizing when just computing their accuracies, but after viewing how they hit and miss, the successes become obvious. That is: the models have not penetrated into the secrets of why these Collatz lengths are what they are, and so they appear to be useless at determining the Collatz length of a number far removed from the training set.

10. **How can the lessons from this study be used to tackle the Collatz problem specifically?**
    The most important takeaway from this study is that if a model appears to be working well on some test set of Collatz data, try again with a test set far removed from the training set. How does the model work then?