

CS EE WORLD

29/34

A

MAY 2018

Topic: Assessing the use of machine learning algorithms for predicting the outcome of random number generators.

Research Question: To what extent can the linear regression machine learning algorithm predict the outcome of the random integer number generator in the Ruby programming language?

Subject: Computer Science

Word Count: 3,985 words

Computer Science Extended Essay

Contents

Introduction.....	2
Background Information.....	4
Machine Learning.....	4
Types of Machine Learning Algorithms.....	4
Linear Regression machine learning Algorithm.....	5
Cost function and Optimization Algorithms.....	6
Gradient Descent Optimization Algorithm.....	7
Figure 1: The Gradient descent algorithm.....	8
Figure 2: Gradient Descent Intuition.....	8
Normal Equation Optimization Algorithm.....	9
Figure 3: Normal Equation Intuition.....	10
Random Number Generator.....	10
Pseudo Random Number Generators (PRNGs).....	11
True Random Number Generators (TRNGs).....	12
Ruby Random Number Generator.....	12
Methodology.....	13
Investigation.....	13
Figure 4: Figure showing how experiment was carried out.....	14
Procedure in Steps.....	15
Data Presentation.....	16
Data Analysis.....	21
Implications of Findings.....	26
Limitations of Investigation.....	27
Conclusion.....	27
References.....	29
Appendix.....	30
Ruby Code Used to generate Random integers.....	30
Linear Regression Algorithm (On Octave) used to train and test data values.....	31

Computer Science Extended Essay

Introduction

Random number generation has been an important part of human life for several years and the methods that are applied to generate these random numbers have evolved due to the development of new uses of random numbers. Random numbers can be generated by physical methods such as die rolling (DiCarlo 4). However, the evolution of the practical uses of random numbers has spearheaded improvement in the procedures used in producing sequences of random numbers.

In recent times, random numbers have begun to be used in government-run lotteries, video games and in modern slot machines. Although the random number generators in some programming languages do not produce true randomness they produce sequences of random numbers that pass standard statistical tests of randomness that measure the unpredictability of the numbers generated. Therefore, since the random numbers are produced using algorithms, there must be a way of finding the pattern of production of random numbers and hence make the values predictable if the right methods are applied.

With increasing development in technology, machine learning has evolved as a popular method of detecting trends in data. Machine learning has numerous applications such as face detection, search engines and weather prediction systems. As such, this investigation seeks to determine the extent to which machine learning can be used as a method of predicting the outcome of random number generators.

Furthermore, random numbers are a crucial component of computations in computers and since they are used in data encryption algorithms (Haahr), their predictability would imply the effortless breaching of security keys by people who can predict the outcome of random number generators. This could possibly lead to a widespread compromise of security systems which would trigger the breaching of the privacy of individuals and corporations.

Computer Science Extended Essay

The aforementioned negative implications of the possible predictability of random number generators show that this investigation is relevant to current society. Usually, a random number generator is easily predictable when the starting value of the sequence of numbers is known. However, this investigation seeks predict the outcome of a random number generator using machine learning even when the initial value is unknown.

Computer Science Extended Essay

Background Information

A random number is a number that forms part of a sequence in which the values are spread uniformly over a defined interval and where there is no possibility of predicting future values based on past or present values (Rouse).

According to the Merriam-Webster dictionary, to predict is defined as “to declare or indicate in advance; especially: foretell on the basis of observation, experience, or scientific reason.”

Machine Learning

Machine Learning is the study of algorithms that are designed to make computers perform functions without human assistance or being explicitly programmed to do so (Stanford University). Machine learning algorithms are made in order to develop applications that can extend their functionality based on example datasets provided to them (Schapire 1). Machine learning forms a crucial part of artificial intelligence and hence is applied in systems related to intelligence such as language or vision. Some other examples of applications of machine learning are face detection programs which find faces in images, spam filtering software that identifies whether a message is spam or not, weather prediction software and search engines (Schapire 1-2).

Types of Machine Learning Algorithms

The various machine learning algorithms are categorized into two main sets namely, supervised and unsupervised learning. The distinction between these two groups is that supervised learning deals with problems where a data set is provided and the correct output is known whereas in unsupervised learning, there is no knowledge or idea of what the result of the computation should look like (Stanford University). Supervised learning algorithms are

Computer Science Extended Essay

further grouped into two groups based on the kinds of problems they solve - classification problems and regression problems.

Classification problems are problems where the given sets of data have to be grouped into defined classifications. The algorithms that deal with classification problems examine the data to detect similarities in the data that will be used in grouping the data (Stanford University). Examples of such machine learning algorithms are the decision tree, the Support, the Naïve Bayes algorithm and the Logistic regression algorithm. (Ray)

On the other hand, the algorithms that are tailored to solve regression problems do not group the given datasets but rather predict results within a continuous output. This is done by establishing a relationship between the input variables and a continuous function (Ray). An example of this is using numerous examples of the land area of houses and their corresponding prices to strike the relationship between these two variables. Then the established relationship is used to determine the price of a house when given its land area (Stanford University). This investigation aims to see how far the outcome of a random number generator can be predicted using a machine learning algorithm. The reason why the linear regression machine learning algorithm was chosen to perform the experiment is because this investigation falls in the category of supervised learning since it involves the prediction of output values based on some input values.

Linear Regression machine learning Algorithm

The Linear Regression machine learning algorithm functions by establishing a general relationship some between independent and dependent variables (Ray). The relationship established is represented by a best fit line which is represented by the linear equation:

$$Y = mX + C$$

Computer Science Extended Essay

Where Y = dependent variable, X = the independent variable, m = the slope of the line and b = y-intercept of the line

The coefficients m and b are determined by implementing a cost function, which will be explained in more detail in the next section. In order to better understand Linear Regression, consider being asked to arrange five books with different sizes in a library according to their weights without using a weighing scale. The most intuitive way to do this would probably be to consider the size of each book, what material it is made up of and possibly it is a hard cover or soft cover book and estimate their weights based on these observations. In the same way the observer establishes a relationship between the features of the book (independent variables) and its weight (dependent variable), the Linear Regression algorithm establishes a relationship between a dependent and an independent variable in the form of a best fit line (Ray).

There are two major kinds of linear regression: the Simple Linear Regression, in which the dependent variable depends on only one independent variable, and the Multiple Linear Regression in which the dependent variable depends on two or more independent variables (Ray). Due to the complex nature of this investigation, it was necessary to implement the Multiple Linear Regression.

Cost function and Optimization Algorithms

The success of linear regression machine learning algorithms depends on their ability to find values of parameters of a function that minimize a cost function (Brownlee). This takes place in a process called training the algorithm where the linear regression algorithm basically learns the pattern of the datasets used for training and calculates values called theta (θ) values with which it can predict the outcome of other datasets that were not used to train the algorithm. During the process of training, the algorithm constantly computes its prediction

Computer Science Extended Essay

(hypothesis) using values of parameters. The algorithm then compares its hypothesis with the actual outcome for each example given to it. The purpose of the cost function during this process is to compute values of parameters which minimize the margin of error between the hypothesis and the actual value for each repetition. These values are conventionally represented with θ . (Stanford University)

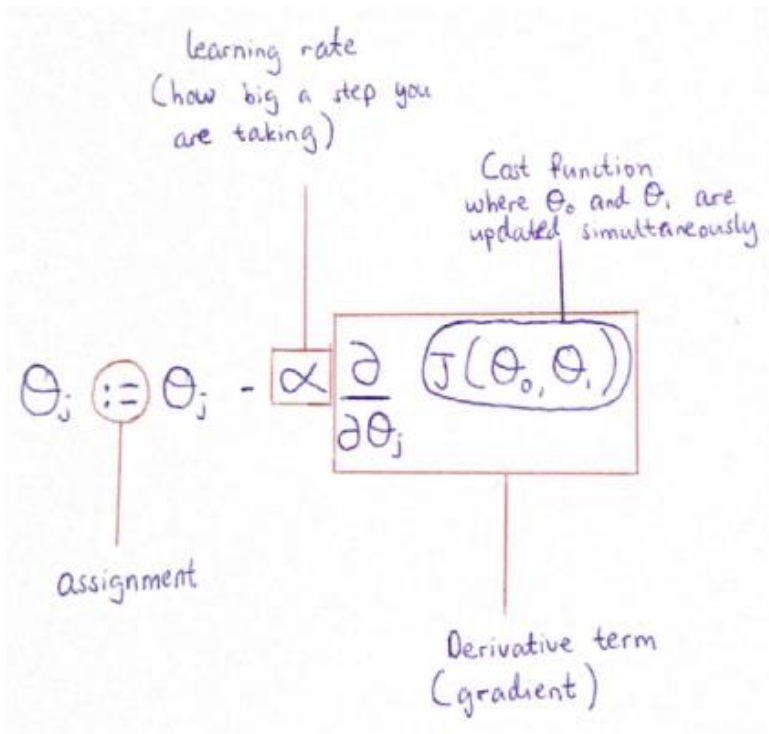
The component of a machine learning algorithm that finds the theta (θ) values is called an optimization algorithm. The role of the optimization algorithm is integral to the machine learning algorithm in that it computes the theta (θ) values which are used to calculate the predicted output. For this investigation, I applied two different optimization algorithms namely, gradient descent and normal equation.

Gradient Descent Optimization Algorithm

The intuition behind gradient descent is easily understood when the training dataset is imagined to be a large bowl which represents the cost function (Brownlee). The bottom of the bowl is considered to be the part with the lowest cost i.e. the part of the cost function that has parameters which yield the least marginal difference between the predicted output value and the actual output value. The function of gradient descent, therefore, is to locate the bottom of the bowl from any starting position (Brownlee). It achieves this by computing the gradient of its current position and then updating the value of θ such that it moves to the next position where the gradient is closer to the minimum point of the bowl – which is a representation of the cost function. This is done by moving in steps – or paces - whose magnitude is set by the person who does the training of the algorithm. The gradient descent algorithm repeats this process until there is convergence i.e. until the bottom of the bowl is reached.

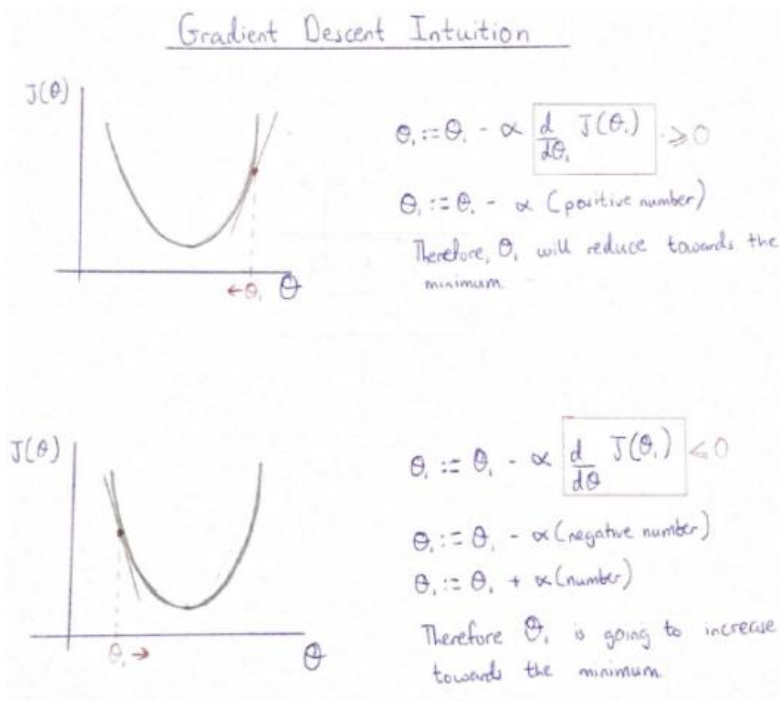
Computer Science Extended Essay

Figure 1: The Gradient descent algorithm



Adapted from (Stanford University)

Figure 2: Gradient Descent Intuition



Adapted from (Stanford University)

Computer Science Extended Essay

Normal Equation Optimization Algorithm

On the other hand, the normal equation algorithm calculates theta (θ) analytically through a matrix multiplication. This is done by grouping the dataset into two matrices, one which contains only the output data and the other which contains the variables that relate to each output. Using an example of the features of a house and its price, the output matrix will contain only the prices of all the houses and the variable matrix will contain characteristics of the house that contribute to the determination of the price of the house such as number of bedrooms, number of bathrooms etc. (Stanford University). The two matrices are then multiplied with an arrangement such that the product of the multiplication will be a one by one (1x1) matrix. This matrix contains the theta (θ) values for each set of features for each house. These theta (θ) values are then used to predict the output values for other sets of data which may not have been used to train the linear regression algorithm. Figure 3 explains how the value of theta is computed using the normal equation algorithm. This optimization algorithm typically takes a shorter time to implement since it is an analytical approach. However, the normal equation algorithm will not work properly for large datasets (about 150,000 independent variables and over) (Stanford University).

Computer Science Extended Essay

Figure 3: Normal Equation Intuition

Normal Equation Intuition

$$X = \begin{matrix} & \text{Feature 1} & \text{Feature 2} & \text{Feature 3} & \text{Feature 4} \\ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} & \begin{bmatrix} 256 \\ 719 \\ 103 \end{bmatrix} & \begin{bmatrix} 10 \\ 12 \\ 15 \end{bmatrix} & \begin{bmatrix} 6 \\ 4 \\ 3 \end{bmatrix} & \begin{bmatrix} 31 \\ 34 \\ 21 \end{bmatrix} \end{matrix}$$
$$y = \begin{matrix} \text{Output} \\ \begin{bmatrix} 7 \\ 9 \\ 15 \end{bmatrix} \end{matrix}$$
$$\Theta = (X^T X)^{-1} X^T y$$

where X^T = transposed version of X

Adapted from

(Ng)

Random Number Generator

According to techopedia.com, “A random number generator (RNG) is a mathematical construct, either computational or as a hardware device, that is designed to generate a random set of numbers that should not display any distinguishable patterns in their appearance or generation, hence the word random.”

As the definition suggests, a random number generator is expected to produce a sequence of numbers that do not display any perceptible patterns or trends within the sequence. Nevertheless there are some types of random number generators that display numbers that have an apparent but synthetic randomness and there are other types that generate numbers with true randomness (DiCarlo 6-7). These random number generators are classified into two main groups: the Pseudo Random Number Generators (PRNGs) the True Random Number Generators (TRNGs).

Computer Science Extended Essay

Pseudo Random Number Generators (PRNGs)

Pseudo random number generators are those that use algorithms that contain mathematical formulae or pre-calculated tables to generate sequences of random number values (Haahr). They generate the sequence of random values by feeding an algorithm with an initial value called the seed value. Some PRNGs allow the seed value to be given either by the user or by the system itself (ruby-doc.org). The algorithm uses the seed value to generate a number which then becomes the seed value for the next computation and this process repeats itself to generate a sequence of seemingly random values (Khan Academy).

Consequentially, pseudo random number generators are not truly random as the name “pseudo” already implies. The values they produce do appear random but are in fact predetermined by the formula or the pre-calculated table of values being used by the generator (Haahr). This characteristic also means that pseudo random number generators are deterministic in the sense that a given sequence random numbers can be regenerated once the seed value of the sequence is given (Haahr). Furthermore, they are periodic meaning that for a number of iterations of producing random number values, the sequence will repeat itself. This may not be a desirable quality to many users of pseudo random numbers but most modern PRNGs possess a period that is long enough be applied for practical uses. (Haahr)

Pseudo random number generators are highly efficient due to the fact that they can produce a large number of random values within a short period of time. This useful quality warrants their application in simulations and modelling software (Haahr). Whereas PRNGs are useful in the aforementioned applications, they are not advised for software that deals with data encryption and gambling due to their predictable nature.

Computer Science Extended Essay

True Random Number Generators (TRNGs)

True random number generators are those that produce sequences of values that have authentic random characteristics. They achieve this by extracting the randomness from physical phenomena that are random in themselves. For example, using data obtained from a radioactive source or atmospheric noise (Haahr). Unlike the PRNGs, the TRNGs are inefficient due to the fact that they take a long time to produce values. They are also non-deterministic and do not have any periods which means that the sequence of random numbers produced does not repeat itself after a number of iterations. Examples of TRNGs are HotBits service at Fourmilab in Switzerland and the lavarand generator built by Silicon Graphics which is no longer in operation. TRNGs are useful for generation of data encryption keys and lotteries and draws where it is crucial that the values used are truly random and cannot be predicted by any means.

Ruby Random Number Generator

Ruby, like other object-oriented programming languages, such as the Java and Python programming languages, utilizes a PRNG to generate sequences of random values. A Mersenne Twister generator is a very popular PRNG. Ruby uses a modified Mersenne Twister generator which has a period of $2^{19937} - 1$. The Ruby PRNG is initialized with either a system-generated seed value or one that is provided by the user and is useful for simulations and modelling applications just like most PRNGs (ruby-doc.org). Apart from the ease with which the ruby programming language allows the user to select the seed for the PRNG, another reason why ruby was chosen for this investigation was due to the fact that it makes use of a PRNG that is common to numerous higher level programming languages. As such, the ruby PRNG stands as a good representative of a standard PRNG which is why it was chosen for this investigation.

Computer Science Extended Essay

Methodology

In carrying out the investigation, it was necessary to firstly generate a set of pseudo-random numbers with the ruby random number generator (see appendix) and store in a Notepad file to make the data accessible to the Linear Regression algorithm –see appendix. The random numbers generated were stored in the arrangement of rows and columns. Then, in order to train the algorithm, a section of the data was uploaded into the linear regression algorithm. After the training process, the remaining section of the data that was not used to train the algorithm was used to test the predictability of the algorithm. The results predicted by the algorithm were then compared with the actual values in order to calculate the error margin between the prediction and actual value. In carrying out this investigation, Komodo IDE was used to implement the Ruby code (see appendix) that produced the random numbers using Ruby's random number generator. The linear regression algorithm – with both gradient descent and normal equation optimization algorithms – was implemented using the GNU Octave which is a programming language used for scientific programming.

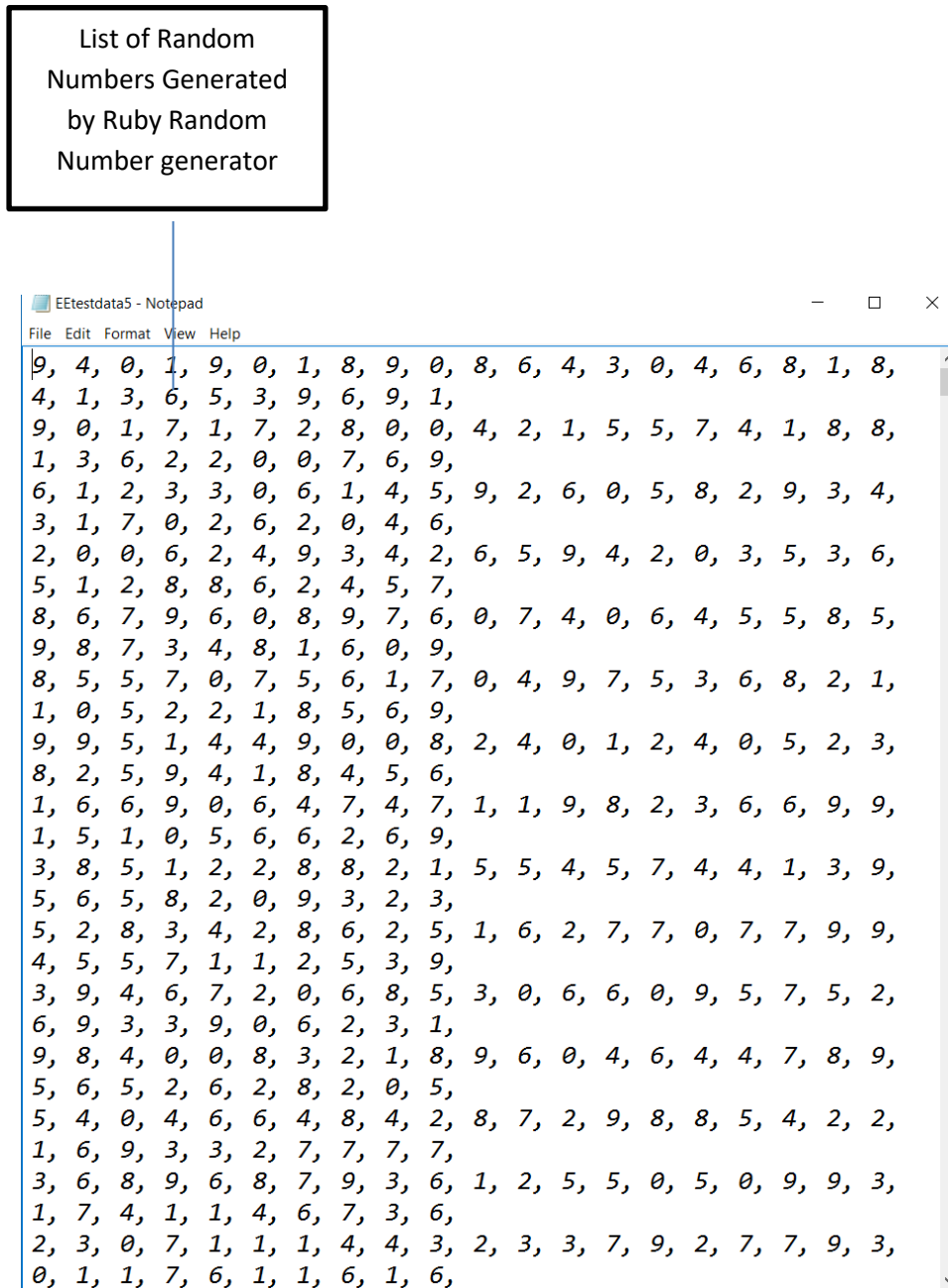
Investigation

To better understand the method used, consider - for instance - a file with 150 rows of random numbers, each with 20 columns. 100 rows of data with 20 columns each would be uploaded into the algorithm for training. Then the remaining 50 rows of random numbers would be used to test the predicting ability of the algorithm after being trained. The testing would be done by inputting a single row of data, which was not used to train the algorithm, up until the 19th column into the algorithm. The algorithm will then be tasked to predict the 20th number of that row. In order to increase the accuracy of my results, the two optimization algorithms – gradient descent and normal equation – were implemented in the Linear Regression machine learning algorithm. This approach was used for my investigation because

Computer Science Extended Essay

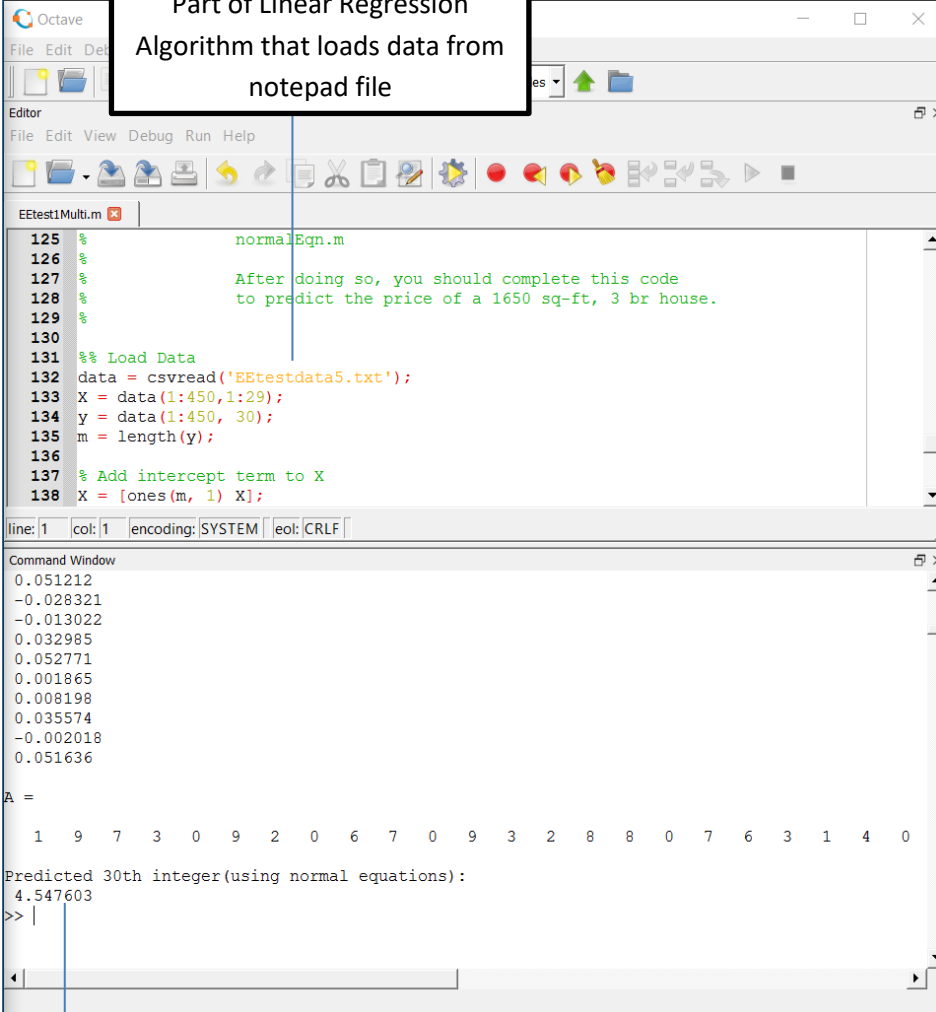
it seemed to be an effective way of testing the patterns derived by the algorithm from the training set of data.

Figure 4: Figure showing how experiment was carried out



Computer Science Extended Essay

Part of Linear Regression
Algorithm that loads data from
notepad file



```
125 % normal Eqn.m
126 %
127 % After doing so, you should complete this code
128 % to predict the price of a 1650 sq-ft, 3 br house.
129 %
130 %
131 %% Load Data
132 data = csvread('EEtestdata5.txt');
133 X = data(1:450,1:29);
134 y = data(1:450, 30);
135 m = length(y);
136 %
137 % Add intercept term to X
138 X = [ones(m, 1) X];
```

Command Window

```
0.051212
-0.028321
-0.013022
0.032985
0.052771
0.001865
0.008198
0.035574
-0.002018
0.051636

A =
  1  9  7  3  0  9  2  0  6  7  0  9  3  2  8  8  0  7  6  3  1  4  0

Predicted 30th integer(using normal equations):
4.547603
>> |
```

Predicted Value 30th number
of a row of random numbers

Procedure in Steps

1. 500 rows and 30 columns of random integers ranging from 0 to 9 were generated using Ruby's pseudo-random number generator (see appendix).
2. 50 rows of random numbers, each with 30 columns, were loaded into the linear regression algorithm for training.
3. Both the gradient descent and normal equation normalization optimization algorithms were applied in the training of the linear regression model.

Computer Science Extended Essay

4. Afterwards, the next fifteen rows from the 50th row of data were used to test the predicting ability of the algorithm. This was done by inputting 29 values of random numbers in a single row into the algorithm while omitting the 30th value, which the algorithm had to predict. Afterwards, the 29 values of the next row of random numbers was input into the algorithm for the prediction of its 30th value. This process was repeated until all the 30th values of all the fifteen rows had been predicted.
5. Steps two to four were repeated, increasing the number of rows of data loaded into the algorithm by 50 with each repetition. With this, the numbers of rows of data used for training were 100, 150, 200, 250, 300, 350, 400 and 450 rows of random numbers from the same Notepad file.

Data Presentation

The data collected from the different numbers of rows used to train the algorithm was organized in tables which show for each row, the predicted values, the actual values, the error margin of each prediction and the average error margin of prediction for the training set.

Computer Science Extended Essay

Table 1: Table Showing the Expected 30th Numbers, their corresponding Predicted Values and the Margins of error for training set of 50 rows for both Gradient Descent and Normal Equation optimization algorithms

		Training Set = 50 rows						
		Gradient Descent			Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin
51	8	2.6353	5.3647		51	8	2.5896	5.4104
52	2	6.9021	4.9021		52	2	7.1375	5.1375
53	2	3.7476	1.7476		53	2	3.6851	1.6851
54	2	7.1434	5.1434		54	2	7.3414	5.3414
55	8	0.6078	7.3922		55	8	0.4495	7.5505
56	6	2.8957	3.1043		56	6	2.5656	3.4344
57	0	7.276	7.276		57	0	7.2378	7.2378
58	9	-2.6993	11.6993		58	9	-2.8716	11.8716
59	2	1.4153	0.5847		59	2	1.438	0.562
60	1	14.0943	13.0943		60	1	14.2164	13.2164
61	4	5.9259	1.9259		61	4	6.2103	2.2103
62	0	6.9269	6.9269		62	0	6.9468	6.9468
63	0	1.8282	1.8282		63	0	1.7478	1.7478
64	3	6.668	3.668		64	3	6.8963	3.8963
65	9	4.5625	4.4375		65	9	4.5476	4.4524
			Ave Error Margin					Ave Error Margin
			5.27300667					5.3800467

Table 2: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 100 rows for both Gradient Descent and Normal Equation optimization algorithms

		Training Set = 100 rows						
		Gradient Descent			Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin
101	2	1.433	0.567		101	2	1.4332	0.5668
102	7	3.5941	3.4059		102	7	3.5925	3.4075
103	8	2.291	5.709		103	8	2.301	5.699
104	1	2.2526	1.2526		104	1	2.248	1.248
105	9	3.6815	5.3185		105	9	3.6781	5.3219
106	4	4.5726	0.5726		106	4	4.5742	0.5742
107	5	3.781	1.219		107	5	3.7786	1.2214
108	4	3.1458	0.8542		108	4	3.1415	0.8585
109	6	2.4163	3.5837		109	6	2.4126	3.5874
110	6	2.4843	3.5157		110	6	2.4811	3.5189
111	2	3.2783	1.2783		111	2	3.2811	1.2811
112	2	3.1466	1.1466		112	2	3.1499	1.1499
113	5	6.5979	1.5979		113	5	6.5955	1.5955
114	7	2.2565	4.7435		114	7	2.2558	4.7442
115	8	5.7102	2.2898		115	8	5.7152	2.2848
			Ave Error Margin					Ave Error Margin
			2.47028667					2.4706067

Computer Science Extended Essay

Table 3: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 150 rows for both Gradient Descent and Normal Equation optimization algorithms

				Training Set = 150 rows					
		Gradient Descent				Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin	
151	4	3.3989	0.6011		151	4	3.3988	0.6012	
152	6	4.4018	1.5982		152	6	4.4019	1.5981	
153	4	5.5592	1.5592		153	4	5.5593	1.5593	
154	2	3.6977	1.6977		154	2	3.6977	1.6977	
155	0	5.4898	5.4898		155	0	5.481	5.481	
56	2	3.5326	1.5326		56	2	3.5326	1.5326	
157	5	1.6184	3.3816		157	5	1.6183	3.3817	
158	7	4.2897	2.7103		158	7	4.2896	2.7104	
159	8	2.2337	5.7663		159	8	2.2337	5.7663	
160	9	3.6117	5.3883		160	9	3.6117	5.3883	
161	3	5.4073	2.4073		161	3	5.4073	2.4073	
162	3	3.9601	0.9601		162	3	3.9691	0.9691	
163	9	2.6766	6.3234		163	9	2.6765	6.3235	
164	2	6.0868	4.0868		164	2	6.0868	4.0868	
165	1	5.004	4.004		165	1	5.0041	4.0041	
			Ave Error Margin					Ave Error Margin	
			3.16711333					3.16716	

Table 4: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 200 rows for both Gradient Descent and Normal Equation optimization algorithms

				Training Set = 200 rows					
		Gradient Descent				Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin	
201	1	5.3734	4.3734		201	1	5.3734	4.3734	
202	7	5.1083	1.8917		202	7	5.1083	1.8917	
203	5	3.6721	1.3279		203	5	3.6722	1.3278	
204	5	5.3651	0.3651		204	5	5.3652	0.3652	
205	5	5.4252	0.4252		205	5	5.4252	0.4252	
206	0	5.8731	5.8731		206	0	5.8729	5.8729	
207	3	4.665	1.665		207	3	4.6649	1.6649	
208	6	4.9658	1.0342		208	6	4.9659	1.0341	
209	3	5.7139	2.7139		209	3	5.7138	2.7138	
210	0	4.7467	4.7467		210	0	4.7465	4.7465	
211	0	5.4851	5.4851		211	0	5.4845	5.4845	
212	9	7.1385	1.8615		212	9	7.1386	1.8614	
213	4	6.1785	2.1785		213	4	6.1786	2.1786	
214	9	4.0043	4.9957		214	9	4.0043	4.9957	
215	8	4.579	3.421		215	8	4.5791	3.4209	
			Ave Error Margin					Ave Error Margin	
			2.82386667					2.8237733	

Computer Science Extended Essay

Table 5: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 250 rows for both Gradient Descent and Normal Equation optimization algorithms

				Training Set = 250 rows					
		Gradient Descent				Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin	
251	4	3.9369	0.0631		251	4	3.9368	0.0632	
252	6	4.7249	1.2751		252	6	4.7249	1.2751	
253	9	5.9176	3.0824		253	9	5.9177	3.0823	
254	8	4.6014	3.3986		254	8	4.6013	3.3987	
255	3	4.375	1.375		255	3	4.375	1.375	
256	9	4.9772	4.0228		256	9	4.9772	4.0228	
257	4	3.7452	0.2548		257	4	3.7451	0.2549	
258	4	3.9095	0.0905		258	4	3.9095	0.0905	
259	0	3.4473	3.4473		259	0	3.4474	3.4474	
260	2	4.2585	2.2585		260	2	4.2585	2.2585	
261	9	3.1274	5.8726		261	9	3.1273	5.8727	
262	2	6.7457	4.7457		262	2	6.7457	4.7457	
263	0	4.9899	4.9899		263	0	4.5899	4.5899	
264	6	6.614	0.614		264	6	6.6141	0.6141	
265	2	6.1301	4.1301		265	2	6.1301	4.1301	
			Ave Error Margin					Ave Error Margin	
			2.64136					2.6147267	

Table 6: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 300 rows for both Gradient Descent and Normal Equation optimization algorithms

				Training Set = 300 rows					
		Gradient Descent				Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin	
301	5	3.899	1.101		301	5	3.899	1.101	
302	6	2.2867	3.7133		302	6	2.2867	3.7133	
303	6	3.7849	2.2151		303	6	3.7849	2.2151	
304	5	4.7642	0.2358		304	5	4.7642	0.2358	
305	2	7.2896	5.2896		305	2	7.2896	5.2896	
306	0	4.3489	4.3489		306	0	4.3489	4.3489	
307	0	4.4517	4.4517		307	0	4.4517	4.4517	
308	2	4.8123	2.8123		308	2	4.8123	2.8123	
309	0	5.5426	5.5426		309	0	5.5426	5.5426	
310	5	4.9231	0.0769		310	5	4.9231	0.0769	
311	3	5.758	2.758		311	3	5.7578	2.7578	
312	3	2.7208	0.2792		312	3	2.7208	0.2792	
313	8	4.1423	3.8577		313	8	4.1423	3.8577	
314	6	5.2332	0.7668		314	6	5.2332	0.7668	
315	6	3.8584	2.1416		315	6	3.8584	2.1416	
			Ave Error Margin					Ave Error Margin	
			2.63936667					2.639353333	

Computer Science Extended Essay

Table 7: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 350 rows for both Gradient Descent and Normal Equation optimization algorithms

				Training Set = 350 rows					
		Gradient Descent				Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin	
351	6	6.4193	0.4193		351	6	6.4193	0.4193	
352	5	4.8322	0.1678		352	5	4.8322	0.1678	
353	4	4.5646	0.5646		353	4	4.5646	0.5646	
354	4	5.1232	1.1232		354	4	5.1232	1.1232	
355	9	3.5393	5.4607		355	9	3.5393	5.4607	
356	7	4.597	2.403		356	7	4.597	2.403	
357	3	4.4045	1.4045		357	3	4.4045	1.4045	
358	6	4.8447	1.1553		358	6	4.8447	1.1553	
359	4	3.198	0.802		359	4	3.198	0.802	
400	2	3.4307	1.4307		400	2	3.4307	1.4307	
401	7	4.7667	2.2333		401	7	4.7667	2.2333	
402	8	3.9744	4.0256		402	8	3.9744	4.0256	
403	4	3.8768	0.1232		403	4	3.8768	0.1232	
404	7	3.3933	3.6067		404	7	3.3933	3.6067	
405	0	4.1087	4.1087		405	0	4.1087	4.1087	
			Ave Error Margin					Ave Error Margin	
			1.93524					1.93524	

Table 8: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 400 rows for both Gradient Descent and Normal Equation optimization algorithms

				Training Set = 400 rows					
		Gradient Descent				Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin	
401	5	4.1837	0.8163		401	5	4.1837	0.8163	
402	3	3.4527	0.4527		402	3	3.4527	0.4527	
403	7	5.1479	1.8521		403	7	5.1479	1.8521	
404	9	3.5131	5.4869		404	9	3.5131	5.4869	
405	0	4.7656	4.7656		405	0	4.7656	4.7656	
406	5	4.0539	0.9461		406	5	4.0539	0.9461	
407	5	5.2019	0.2019		407	5	5.2019	0.2019	
408	1	2.8841	1.8841		408	1	2.8841	1.8841	
409	4	3.9377	0.0623		409	4	3.9377	0.0623	
410	6	5.9239	0.0761		410	6	5.9239	0.0761	
411	4	4.4036	0.4036		411	4	4.4036	0.4036	
412	4	4.3616	0.3616		412	4	4.3616	0.3616	
413	5	4.3153	0.6847		413	5	4.3153	0.6847	
414	0	3.4709	3.4709		414	0	3.4709	3.4709	
415	2	4.6702	2.6702		415	2	4.6702	2.6702	
			Ave Error Margin					Ave Error Margin	
			1.60900667					1.60900667	

Computer Science Extended Essay

Table 9: Table Showing the Expected 30th Number, their corresponding Predicted Value and the Margin of error for training set of 450 rows for both Gradient Descent and Normal Equation optimization algorithms

				Training Set = 450 rows					
		Gradient Descent				Normal Equation			
Row Number	Expected 30th Number	Predicted 30th Number	Error margin		Row Number	Expected 30th Number	Predicted 30th Number	Error margin	
451	1	3.7478	2.7478		451	1	3.7478	2.7478	
452	6	5.4145	0.5855		452	6	5.4145	0.5855	
453	8	5.0676	2.9324		453	8	5.0676	2.9324	
454	8	3.9021	4.0979		454	8	3.9021	4.0979	
455	0	4.5364	4.5364		455	0	4.5364	4.5364	
456	7	4.8678	2.1322		456	7	4.8678	2.1322	
457	0	4.614	4.614		457	0	4.614	4.614	
458	7	4.2783	2.7217		458	7	4.2783	2.7217	
459	5	4.5174	0.4826		459	5	4.5174	0.4826	
460	1	4.5476	3.5476		460	1	4.5476	3.5476	
461	2	4.0686	2.0686		461	2	4.0686	2.0686	
462	3	4.9959	1.9959		462	3	4.9959	1.9959	
463	0	3.8382	3.8382		463	0	3.8382	3.8382	
464	2	5.4757	3.4757		464	2	5.4757	3.4757	
465	3	4.4457	1.4457		465	3	4.4457	1.4457	
			Ave Error Margin					Ave Error Margin	
			2.74814667					2.74814667	

Data Analysis

In analysing the data presented, Microsoft Excel was used to calculate the average error margin of predictions of each training set and attempted to find a relationship between the error margin of predictions and the number of rows used as the training set for the algorithm. I did this as a step towards finding the least possible margin of error in each case. This is because finding the least possible margin of error would be a step closer to finding a conclusion to the investigation since it seeks to find the extent to which random integers can be predicted by using linear regression as a machine learning algorithm.

Computer Science Extended Essay

Table 10: Table of Average Error Margins and the Number of Rows of Data used as the training set

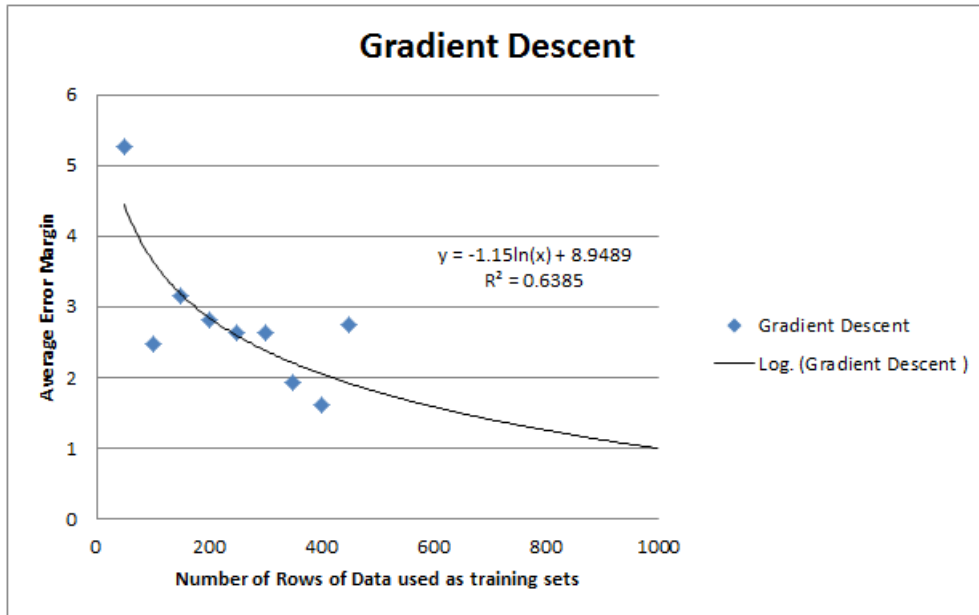
Gradient Descent			Normal Equation	
Number of Rows used as training sets	Average Error Margin		Number of Rows	Average Error Margin
50	5.273006667		50	5.380046667
100	2.470286667		100	2.470606667
150	3.167113333		150	3.16716
200	2.823866667		200	2.823773333
250	2.64136		250	2.614726667
300	2.63936667		300	2.6393533
350	1.93524		350	1.93524
400	1.60900667		400	1.60900667
450	2.74814667		450	2.74814667

The data shown in Table 2 are the combined results of my experimentation with the various sizes of datasets used for training and how the average error margin of prediction varied as the size of the training set used was increased. The reason for the large number of decimal places in the average margin of error values is to achieve as much precision as possible in the analysis of the data. A general trend is seen from the table in that, with an increase in the number of rows of random number used for training, the average margin of error of the predictions decreased. In order to further analyse these results, I represented the data in two graphs, one for the results achieved using gradient descent and the other for the results achieved using the normal equation approach.

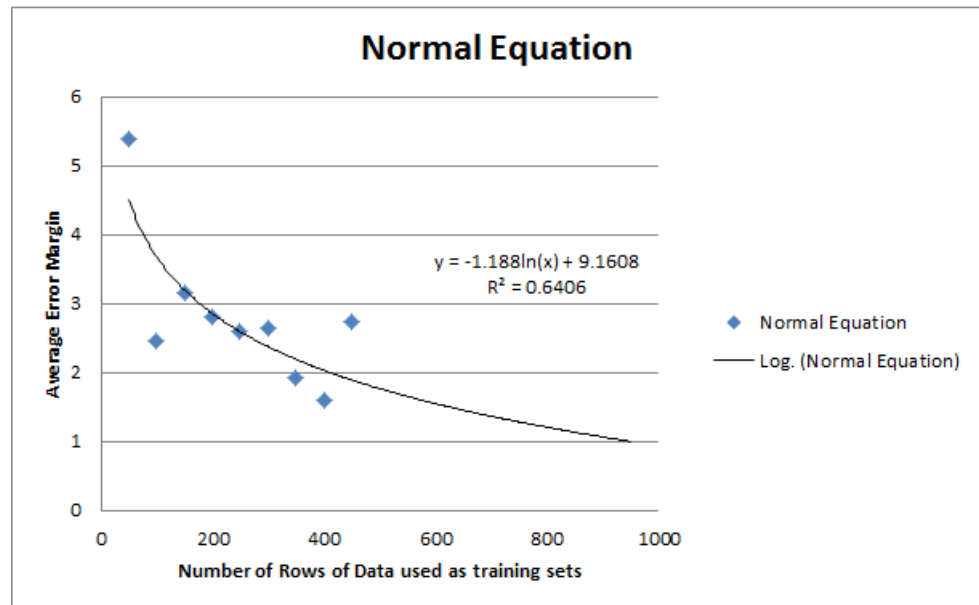
The two following graphs represent this data in a better way for a trend to be seen in the results.

Computer Science Extended Essay

Graph 1: A Graph of Average Error Margin against the Number of Rows of Data tested using Gradient Descent



Graph 2: A Graph of Average Error Margin against the Number of Rows of Data tested using Normal Equation

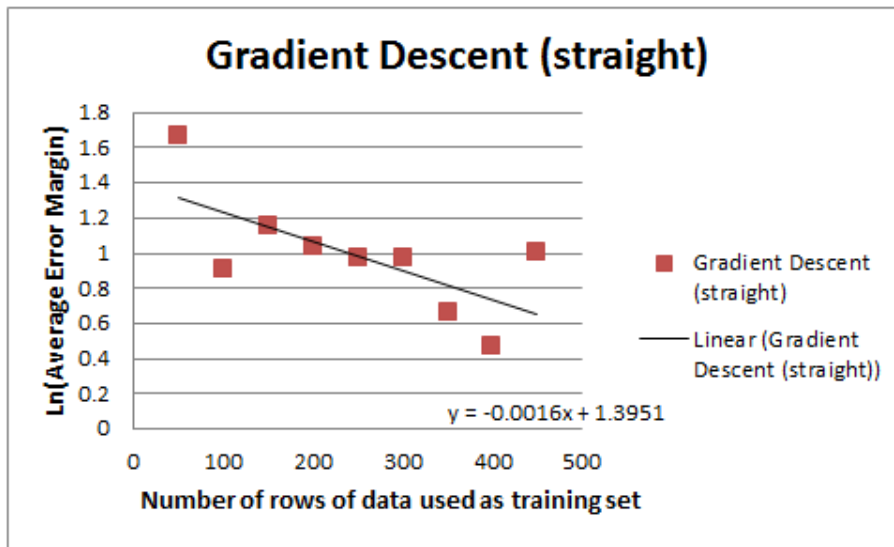


The graphs for both the gradient descent and normal equation optimisation algorithms show that the relationship between the average error margin and the number of rows of data used as the training set is logarithmic – which is not quite helpful for a clear analysis to be done. In

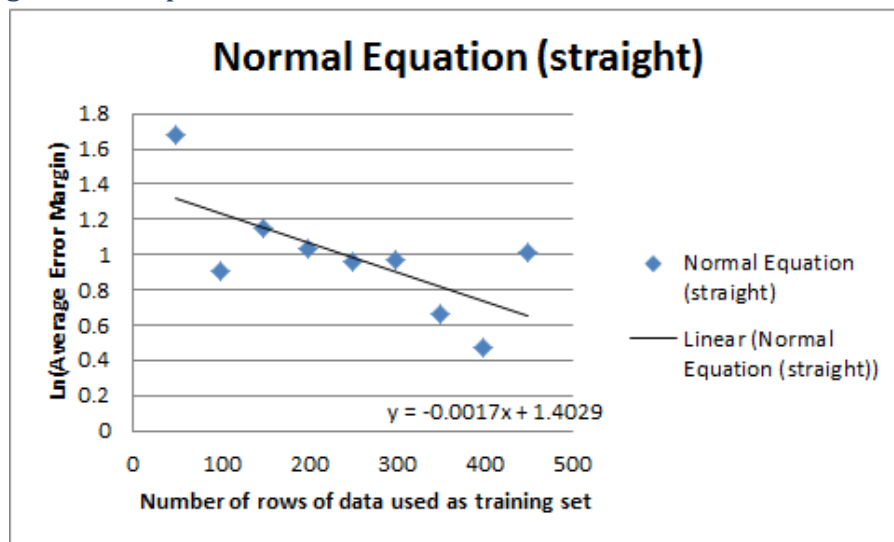
Computer Science Extended Essay

order to better analyse the relationship from these graphs, they were converted into a straight-line graphs by plotting the Ln (Ave. error margin) against the number of rows used as the training set. The following graphs demonstrate this.

Graph 3: A Graph of Ln (Average Error Margin) against the Number of Rows of Data tested using Gradient Descent



Graph 4: A Graph of Ln (Average Error Margin) against the Number of Rows of Data tested using Normal Equation



Computer Science Extended Essay

The straight line graph shown by the results achieved by Gradient Descent has the equation,

$$y = -0.0016x + 1.3951$$

That of Normal Equation has the equation

$$y = -0.0017x + 1.4029$$

From these equations, a relationship between the average error margin between predictions of the random numbers and their corresponding expected values, and the number of rows of data used to train the linear regression algorithm can be derived.

Equation derived from the equation of the graph for Gradient Descent:

$$\ln(\text{Ave. Error Margin}) = -0.0016(x) + 1.3951$$

$$\text{Ave. Error Margin} = e^{-0.0016x+1.3951}$$

Where x = the number of rows used as a training set

Equation derived from the equation of the graph for Norman Equation:

$$\ln(\text{Ave. Error Margin}) = -0.0017(x) + 1.4029$$

$$\text{Ave. Error Margin} = e^{-0.0017x+1.4029}$$

Where x = the number of rows used as a training set

The minimum values of these two equations are 1.9752 and 1.90373 respectively.

Computer Science Extended Essay

These findings suggest that the least average margin of error that can be achieved by the linear regression machine learning algorithm in attempting to predict the outcome of the ruby random integer number generator is approximately 2. This value represents the extent to which the random integer can be predicted by the linear regression algorithm and therefore answers the research question. This value suggests that although the prediction of the random integer may not be the exact value of the actual outcome of the random number generator, it is quite close to it. An interesting factor to take notice of is the fact that the average margin of error decreased significantly from approximately 5 to approximately 2 as the number of rows of data used for training was increased from 50 rows to 450 rows.

Implications of Findings

The findings of this investigation imply that machine learning could be a plausible way of predicting the outcomes of pseudo-random number generators, given a sufficient amount of data for training. This could suggest the reason why systems that demand high security such as gambling systems and encryption algorithms do not use of PRNGs. Conversely, the findings also suggest that the linear regression machine learning algorithm allows enough accuracy to be applied in fields that require less security. One such field is the prediction of grades in IB. As seen in anonymous data provided by the school showing past students' predicted grades and actual grades, the average margin of error calculated was 2. As a similar margin of error was produced by the linear regression model, from the investigations findings, it suggests that one useful application of the linear regression machine learning algorithm could be that of the prediction of students' grades. In this instance, the linear regression algorithm could be trained with data of the grades of graduated students from their IB1 first semester exam to their final IB exam. After the linear regression algorithm is trained, it can then be fed with the grades of current students from their IB1 first semester

Computer Science Extended Essay

exam to the last exam they write before the final exam. The algorithm can then be made to predict the final IB grade of the student based on the trends found from the training sets. The prediction of IB grades using the linear regression algorithm is also considered to be a plausible idea due to the small range of possible IB grades – 1 to 7. Due to this small range, the pattern of the grades achieved by an IB student might not be hard to find as opposed to having a larger range to work with.

Limitations of Investigation

The limitations of the investigation include the following:

1. The range of numbers generated by the pseudo-random number generator was small. The data experimented on consisted of random integers ranging from 0 to 9. This is a limitation as only single digit integers were considered whereas in practice, multi-digit random integers are often used. This is due to the fact that a higher range of digits would make the pattern of random integers generated harder to find, thereby, increasing their secure nature.
2. Another limitation of the investigation stems from the fact that only random integers were considered during the experimentation process. This investigation did not consider the predictability of random decimal numbers. Typically, random decimal numbers would render the pattern of random numbers more difficult to predict since there will be more permutations of figures available to form the pattern.

Conclusion

The investigation sought to find the extent to which the linear regression machine learning algorithm can predict the outcome of the random integer generator in the ruby programming language. At the end of the process of experimentation and analysis of results, it can be

Computer Science Extended Essay

concluded that linear regression as a machine learning algorithm can be used to predict outcomes of the random integer generator with an average margin of error of 2. This conclusion stands as reliable despite all the limitations of this investigation since the linear regression algorithm can be modified to predict the outcomes of random sequences with more complex patterns. This investigation only serves as a demonstration of the vast possibilities of prediction using linear regression as a machine learning algorithm.

Computer Science Extended Essay

References

Britt, James, and Neurogami. "Class: Random (Ruby 2.2.0)." *Ruby-doc.org*. Web. 18 Aug. 2017.

Brownlee, Jason. "Gradient Descent For Machine Learning." *Machine Learning Mastery*. N.p., 2016. Web. 22 Oct. 2017.

DiCarlo, David. *Running Head: Random Number Generation*. Lynchburg: Liberty University, 2012. Web. 27 Oct. 2017.

Haahr, Mads. "RANDOM.ORG - Introduction To Randomness And Random Numbers." *Random.org*. Web. 18 Aug. 2017.

KhanAcademy. *Pseudorandom Number Generators*. 2012. Web. 18 Aug. 2017.

Ng, Ritchie. *This Image Is An Annotated Diagram Of The Normal Equation Intuition*. 2017. Web. 15 July 2017.

"Predict." *Merriam-Webster* 2017. Web. 18 Aug. 2017.

"Random Number Generator." *Technopedia* 2017. Web. 18 Aug. 2017.

Ray, Sunil. "Essentials Of Machine Learning Algorithms (With Python And R Codes)." *Analytics Vidhya*. N.p., 2015. Web. 14 May 2017.

Rouse, Margaret. "What Is Random Numbers? - Definition From Whatis.Com." *WhatIs.com*. N.p., 2005. Web. 18 Aug. 2017.

Computer Science Extended Essay

Schapire, Rob. *COS 511: Theoretical Machine Learning*. New Jersey: Rob Schapire, 2008.

Web. 18 Aug. 2017.

Stanford University. *Classification*. Web. 1 July 2017.

Stanford University. *Gradient Descent Intuition*. 2017. Web. 17 July 2017.

Stanford University. *Supervised Learning*. Web. 7 July 2017.

Stanford University. *Unsupervised Learning*. Web. 8 July 2017.

Appendix

Ruby Code Used to generate Random integers

```
#!/usr/bin/env ruby
seed = 10
count = 0
while count < 500
  prng = Random.new(seed)
  30.times{print "#{prng.rand(10)}, "}
  seed +=1
  count +=1
  puts ""
end
```

Source: Author

Computer Science Extended Essay

Linear Regression Algorithm (On Octave) used to train and test data values

```
%% ===== Part 1: Feature Normalization =====

clear ; close all; clc

%
data = csvread('EEtestdata5.txt');
X = data(1:450,1:29);
y = data(1:450, 30);
m = length(y);

% Scale features and set them to zero mean
fprintf('Normalizing Features ...\n');

[X mu sigma] = EEfeatureNormalize(X);

% Add intercept term to X
X = [ones(m, 1) X];

%% ===== Part 2: Gradient Descent =====
fprintf('Running gradient descent ...\n');

alpha = 0.001;
num_iters = 20000;

% Init Theta and Run Gradient Descent
theta = zeros(30, 1);
[theta, J_history] = EEgradientDescentMulti(X, y, theta, alpha, num_iters);

% Plot the convergence graph
figure;
plot(1:numel(J_history), J_history, '-b', 'LineWidth', 2);
xlabel('Number of iterations');
ylabel('Cost J');

% Calculate the parameters from the normal equation
theta = EEnormalEquation(X, y);

% Display normal equation's result
fprintf('Theta computed from the normal equations: \n');
fprintf(' %f \n', theta);
fprintf('\n');

% Estimate the 30th number of random sequence
A = data(R,1:29);
A = [ones(1,1) A]
thirtieth_number = A *theta;
fprintf(['Predicted 30th integer' ...
        '(using normal equations):\n %f\n'], thirtieth_number);
```

Adapted from (Stanford University)

Computer Science Extended Essay

Software Used

Software Used to generate random numbers: Komodo IDE 11

Software used to implement linear regression algorithm: GNU Octave