Name(s)
Jack D. Albright

Project Number
J1501

Project Title
Predicting the Future: Using Machine Learning to Forecast the Progression of Alzheimer's Disease

Abstract

Objectives/Goals
Alzheimer's disease (AD) is the most common neurodegenerative disease in older people. Despite considerable efforts to find a cure for AD, there is a 99.6% failure rate of clinical trials for AD drugs, likely because AD patients cannot easily be identified at early stages. AD research would benefit from the ability to use current patient data to predict the clinical state of patients in future years. This project investigated machine learning approaches to enable such predictions.

Methods/Materials
Clinical data from 1737 patients was obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database and was preprocessed using a novel methodology involving the comparison of all possible pairs of temporal data points for each patient. This data was then used to train machine learning models implemented with the Python library scikit-learn. Models were first evaluated using k-fold cross-validation on the training dataset, and then confirmed using data from a separate testing dataset (110 patients).

Results
Several machine learning-based classifiers, including support vector machines and neural networks, were effective (86.6% MAUC score) at predicting the progression of AD, both in patients who were initially cognitively normal and in patients suffering from mild cognitive impairment.

Conclusions/Discussion
This project developed a machine learning model that can correlate clinical data obtained from patients at one time point with the month-by-month progression of AD in the future. Such a model could be used to identify patients having high AD risk before they become symptomatic and who are therefore good candidates for clinical trials for AD therapeutics.

Summary Statement
I developed a machine learning model and novel preprocessing technique that uses clinical data to identify patients at risk for Alzheimer's disease and predict the future progression of their clinical diagnoses on a month-by-month basis.

Help Received
I received guidance from my computer science teacher Jen Selby and my science teacher Rachel Dragos. I also received feedback on my machine learning model from Alex Abbas, a bioinformatics researcher at Genentech, and Lisa Bell, a regulatory consultant experienced in neurological disease clinical trials.

CALIFORNIA SCIENCE & ENGINEERING FAIR
2018 PROJECT SUMMARY

Name(s)
Brady M. Elliott

Project Number
J1502

Project Title
All in the Numbers: A Mathematical Analysis of Current Boston Marathon Qualifying Standards

Abstract

Objectives/Goals
The objective and goal of my science project was to test if the Boston Marathon Qualifying Standards are accurate for both male and female participants of all ages. The question that I formulated was; are the current qualifying standards to run the Boston Marathon an accurate reflection of the times being run by the top age group athletes completing in four largest marathons in the United States?

Methods/Materials
Method- 1.Go to the official websites for the New York, Los Angeles, and Marine Corps Marathons 2. Calculate the top 5% of each age group participants. 3. Use ruler to make a scatter plot graph to plot participants.Materials- 1. 2019 Boston Marathon Qualifying Standards 2. Marathon Participants race results 3. graph paper 4. straight edge

Results
I formulated that the middle aged participants age 21 to 74 for both male and female are an accurate reflection of the times being run by all of the participants in this age group. However, I was correct since the younger and older aged participants were inaccurate. I formulated that the 18 to 20 age group time should be lowered and the 75+ age group should be lowered.In fact, the under 20 male runners had a qualifying standard where only 12 runners qualified.

Conclusions/Discussion
My science fair project was to test if the current Boston Marathon qualifying standards are an accurate reflection of the times being run by the top 5% of the age group runners. My hypothesis, based on my research, was that the qualifying standards to run the Boston Marathon are inaccurate for male and females of all ages. My hypothesis was proven partially incorrect after graphing the times of the Boston Marathon participants. I was proven partially incorrect since the middle aged participants age 21 to 74 for both male and female are an accurate reflection of the times being run by the participants in this section of age.I was correct since the younger and older aged participants were inaccurate. I formulated that the 18 to 20 age group time should be lowered and the 75+ age group should be lowered. In fact, the under 20 male runners had a qualifying standard where only 12 runners qualified, and 45 runners failed to qualify.Lastly, this project could help humanity since many people are questioning the accuracy of the Boston Marathon qualifying standards without doing the research. My project could help conclude the complaining and show the true accuracy of the Boston Marathon.

Summary Statement
I proved the inaccuracy of younger and older participant qualifying times, while proving the middle aged qualifying standards to be an accurate reflection of the times being run.

Help Received
Mrs. Rodriguez:Science teacher Mrs. Elliott: English teacher Mr. Elliott: Math teacher

Name(s)
John H. Gerving

Project Number
J1503

Project Title
Detecting Cancer with Neural Networks

Abstract

Objectives/Goals
With the advent and availability of large quantities of data, powerful artificial intelligence models are becoming increasingly more popular for use by the general public. The goal of this project was to develop a tool that could be used in the medical field to identify malignant and benign cells. Currently, scientists look at cells under a slide and use their training to identify a cell. This process could be mimicked with an artificial neural network (ANN). An ANN, similar to a human brain, takes an input and multiplies it by a random value to produce a prediction. Depending upon how far off that prediction is, the network adjusts those values to better suit the situation. This process is considered "learning." The objective was to see if an ANN could correctly differentiate between malignant and benign cells, and create a mobile app containing the neural network that people could download onto their phones.

Methods/Materials
After conducting research, it was discovered that it would take too much computing power to create a custom image recognition ANN. Instead, an open-source ANN made by Google called Inception was used. Dr. Megan Smith-Zagone, a pathologist at St. Joseph's Hospital, provided pictures of malignant and benign cells for use in the project. The Inception model was trained on 50 pictures each of malignant and benign cells and was tested on new pictures, with each experiment increasing the number of times the network trained itself.

Results
After being trained 4200 times, the network achieved 96% training accuracy and 100% testing accuracy. Training accuracy is the accuracy determined by the network testing itself and calculating its accuracy on its own. Testing accuracy is the accuracy determined by a human training it on new images. The reason the network achieved "100%" testing accuracy is because there were only 10 images to test the network. If more testing images had been used, the testing accuracy would likely have been closer to 96%.

Conclusions/Discussion
A program was created that could correctly identify malignant and benign cells. Sometimes, scientists are unsure whether cells are malignant or benign, so this program could help make this task easier. In addition, the Inception model was successfully deployed onto an Android phone, so this app could potentially be used to identify malignant and benign cells easily in a lab.

Summary Statement
In this study, I was able to teach a neural network how to identify malignant and benign cells using pre-acquired images.

Help Received
I adapted a pre-made neural network made by Google. Dr. Megan Smith-Zagone provided pictures of malignant and benign cells for use in training the neural network.

CALIFORNIA SCIENCE & ENGINEERING FAIR
2018 PROJECT SUMMARY

Name(s)
Su Kara

Project Number
J1504

Project Title
Proof of Pappus Theorem with Circle Inversion by Developing an Open Source Software Application

Abstract

Objectives/Goals
Develop an open-source software application to simulate circle inversion and prove Pappus' theorem.

Methods/Materials
MacBook Pro to develop a web page in HTML5 and JavaScript. Wrote the source code in Brackets, an open-source text editor, to invert a point, circle, and Pappus chain. Tested and debugged the software in Safari.

Results
A web page that displays three tabs to invert a point, to invert a circle, and to create a Pappus chain, invert its parts, and show homothety. The user can either load a predefined template, or enter custom values to run their own inversions.

Conclusions/Discussion
I created a web page with graphics features to simulate circle inversion. I proved Pappus theorem' by visually showing similarity between circles in the Pappus chain and their inversions. It runs in the latest browsers on all computers and mobile devices. I share this tool as an open-source software application with anyone interested in math, and specifically in circle inversion and Pappus chain. Please feel free to use the application and get the source code from my web page at http://sukarablog.weebly.com.

Summary Statement
I developed a web application to prove the Pappus theorem by simulating circle inversion.

Help Received
I designed, developed, and tested the program myself.

Name(s)
Colin S. Kovarik

Project Number
J1505

Project Title
Comparing the Efficiency of Popular Pathfinding Algorithms on Random Networks

Abstract

Objectives/Goals
Pathfinding programs are on of the most applied algorithms in present day. In my project, I will be coding four pathfinding artificial intelligence programs (A*, Uniform Cost, Breadth-first and Depth-first search) to compare the efficiency of the algorithms based on the number of iterations and the length of the path on randomly generated networks. I believe that depth first search will be the most efficient, but also least accurate. I hypothesize that Uniform Cost will always find the shortest path, but A* will find a longer path but with a better path length and iteration ratio than Uniform Cost. I think Breadth-first will be the least efficient with the highest iteration to path ratio, and will only find a slightly shorter path then depth first search.

Methods/Materials
First, I designed and coded a program to generate random networks with 100 to 25000 points. I then coded 4 different pathfinding methods: A*, Uniform Cost, Breadth-first and Depth-first. I ran and repeated each program at least 1000 on different randomly generated networks. Additionally, I tested A* search with different heuristics.

Results
I found that Uniform Cost consistently found the shortest path. Breadth-First search average path found was 1297 units, Depth-first search had a average of 2462 units and Uniform Cost typical path was at 852. With a heuristic of 1, A*1 search had a typical path of 892. When the heuristic was set to 1.5, A*1.5 search averaged 923 units. And at a heuristic of 2, A*2 search averaged 944 units. Although Depth-first search resulted in the greatest average distance, it had the lowest Iterations to Distance ratio at 0.03. A* search was the second most efficient algorithm, with A*2 averaging 0.13, A*1.5 averaging at 0.174, and A*1 averaging at .47. Uniform cost averaged at 3.47, followed by Breadth-first at 278.99.

Conclusions/Discussion
My hypothesis is supported by my results. Uniform cost finds the shortest path, however it has a 3529% increase in iterations compared to depth-first search. However, the path depth-first finds is 189% larger than the one Uniform cost finds. A* search varied with different heuristics, but averaged at a 190% increase in iterations and 8% increase in pathlength. The difference between the algorithms grew exponentially. Breadth-first search was the least efficient method with a 52% percent increase in path length, and a 4348.69% increase in iterations.

Summary Statement
Comparing the number of iterations needed to find a path on random networks.

Help Received
None. I designed and coded the experiments myself.

CALIFORNIA SCIENCE & ENGINEERING FAIR
2018 PROJECT SUMMARY

Name(s)
Sanskriti Singh

Project Number
J1506

Project Title
Happy Friend or Angry Stranger? A Neural Network Based Smart and Affordable Assistant System for Visually Impaired Peopl

Abstract

Objectives/Goals
There is a need for a computer vision based assistant system for visually impaired people to help them recognize people, their emotion and maybe to navigate them through the streets. I want to build a Raspberry Pi based camera system running state of the art artificial intelligence face and emotion recognition software written in python.

Methods/Materials
First, finding efficient open source python library to find face locations and recognize faces. Study and implement feed forward neural network in python to predict emotions. Attach camera, touch screen and speaker to Raspberry Pi, implement the code to take picture when instructed, locate faces in the picture, run face recognition software to identify faces, recognize people, and their emotions, combine all information and output to speaker using text to speech library. System will say [Happy|Sad|Neutral|Angry|Disgust|Fear|Contempt|Surprise][<Name of the person>|Stranger].

Results
I was successfully able to implement and integrate the system, which takes the picture, locate faces and identify people and their emotions. It displays the picture with additional information added and speak the information. It recognizes people with 94% accuracy and predicted emotions with 70% accuracy.

Conclusions/Discussion
My project successfully demonstrated the concept of such an assistant device. Accuracy of emotion recognition can be improved further by the use of CNN. Additional computer vision functions can be added to identify obstacles in the path of people and to navigate them.

Summary Statement
Computer vision based smart and affordable assistant system for visually impaired people.

Help Received
Mr. Manish Singh, Principal Engineer @Ambarella

Name(s)
Espen Slettnes

Project Number
J1507

Project Title
Minimal Embedding Dimensions of Rectangle k-Visibility Graphs

Abstract

Objectives/Goals

Research on bar visibility graphs was originally motivated by problems about constructing VLSI (Very Large Scale Integration) circuits, and were adopted in the 1980s as a geometric model to represent traces, e.g. on circuit boards and in VLSI chip designs.

Rectangle visibility graphs were introduced by Bose et al in 1997 as a generalization of bar visibility graphs. A graph is a rectangle visibility graph if it can be represented with vertices as disjoint axis-parallel rectangles, such that there is an unobstructed axis-parallel line of sight between two rectangles if and only if there is an edge between the corresponding vertices.

I combined rectangle visibility graphs with k-visibility to form rectangle k-visibility graphs, in which the line of sight between two rectangles in the representation can be obstructed by at most k other rectangles.

I then took a natural generalization of rectangle k-visibility graphs into higher dimensions. I found that given enough spacial dimensions there exists a rectangle k-visibility representation of any graph G. I continued to study its properties, and proceeded to bound it for complete graphs, complete r-partite graphs, and hypercube graphs.

Results

I established upper bounds on the number of dimensions needed to represent the above types of graphs as rectangle k-visibility graphs, in some cases with the added restriction that the rectangles be unit rectangles, and/or that k=0. Additionally I established a similar upper bound on the minimal embedding dimension on the Cartesian product of multiple graphs.

Conclusions/Discussion

The representation of graphs as hyper-rectangles with k-visibility lines is an exciting extension of existing visibility graph concepts, and like previous work in the field, is likely to have applications not yet imagined. In future research I hope to sharpen the bounds presented here, to study additional types of graphs, and to study different types of visibility.

Summary Statement

I explored the number of dimensions required to represent various graphs as hyperrectangles with axis-parallel visibility lines.

Help Received

Through a MIT PRIMES-USA research internship, I received mentorship from Dr. Jesse Geneson, who introduced me to the topic of visibility graphs.

CALIFORNIA SCIENCE & ENGINEERING FAIR
2018 PROJECT SUMMARY

Name(s)
Mahit V. Tanikella

Project Number
J1508

Project Title
Autism Spectrum Disorder Screening Using Machine Learning

Abstract

Objectives/Goals
Autistic Spectrum Disorder (ASD) is a neurodevelopment disorder characterized by impaired communication, cognitive, and social skills and abilities. Existing screening tools for detection of autism are expensive, cumbersome and time-intensive. The objective of this project is to create a low cost, quick and easy to use diagnostic test for ASD by building a machine learning algorithm that can predict with close to 100% accuracy, whether a person has ASD, based on behavioral traits.

Methods/Materials
Machine learning models were developed for five different classification algorithms namely Logistic Regression (LR), Decision Trees (DT), Gaussian Naive Bayes (NB), Support Vector Machines (SVM) and Neural Networks (NN). Coding was done in Python using scikit-learn in Jupyter notebook. The models were trained using ASD screening data from UC Irvine machine learning repository. Data consists of response to questions on behavioral traits, age, gender, ethnicity, family history and if the person had jaundice when born. For evaluating the models, 10-fold cross validation technique was used in which the data is partitioned into ten equal sizes and nine samples were used for training and one for validation. Accuracy score, confusion matrix that describes performance of the model and classification report were generated for each model.

Results
The models developed have achieved average accuracies of 96.7%(LR), 90.7%(DT), 95.4%(NB), 92.3%(SVM) and 97.4%(NN) with standard deviations of 0.023(LR), 0.036(DT), 0.034(NB), 0.035(SVM) and 0.01(NN) respectively. Neural networks based model is the best with highest possible accuracy and lowest variance. Dropping gender and age from the input feature list improved accuracy which means they are not useful for predicting ASD. Accuracy of models drop if only response to questions i.e. behavioral traits are used for training. Family history, ethnicity and if the person was born with jaundice are important factors to consider along with behavioral traits for ASD screening.

Conclusions/Discussion
Neural networks based machine learning model developed predicts if someone has ASD with 97.4% accuracy, based on answers to behavioral traits questions. People can take this test from the comfort of their home, on their computer or mobile phone for initial assessment, before doing more expensive diagnostic tests.

Summary Statement
Neural networks based machine learning model developed predicts if someone has Autism Spectrum Disorder with 97.4% accuracy, based on answers to behavioral traits questions.

Help Received
My mom helped me with questions I had when I did the online course to learn how to use scikit-learn machine learning library.

Name(s)
Alexa A. Wingate

Project Number
J1509

Project Title
Fermat vs. Brute Force: The Making of a Better Primality Test

Abstract

Objectives/Goals
RSA encryption is used to protect a person's privacy on the internet. Part of the encryption process involves quickly distinguishing large prime and composite numbers. The objective of this experiment is to build a much more efficient primality test using probabilistic methods.

Methods/Materials
For this experiment, I programmed my own Probabilistic Fermat primality test using python and also built a simple deterministic brute force algorithm to compare it against. To exploit Fermat's unique advantages to vary confidence levels, I introduced a "Fermat Witness Limit" parameter to control the number of iterations that the Fermat test performed. With this, I could trade-off precision for speed with the intention of discovering the right balance between the two goals.

To compare the Probabilistic approach (Fermat) vs the deterministic (brute force) approach, my program ran many numbers through both engines while monitoring the time it took for each of the methods to determine a number's primeness. Then I also added an error-checking component to my python program to see how varying the "Fermat Witness Limit" would add imprecision. I optimized the engine by testing multiple scenarios against the deterministic (brute force) approach. The data was exported to excel to display the relative performance of the two distinct approaches.

Results
The results of this experiment found that as the numbers got larger, the Fermat primality test could decrease the Fermat Witness Limit without sacrificing much imprecision. This would make it find prime numbers the size that would be used in RSA encryption up to 10^100 times faster than the brute force algorithm. However, the Fermat approach was only equally as good compared to the brute force algorithm at verifying composite numbers.

Conclusions/Discussion
The results of this experiment fulfill the objective of the experiment in creating an efficient algorithm for finding primes. The Fermat primality test was found to be much, much faster than the brute force algorithm for finding primes, although it was no better at identifying composites.

Summary Statement
I created an efficient primality test using a probabilistic approach and Fermat's Little Theorem.

Help Received
My former mentor, David Crane, helped me to make the Fermat primality test more efficient by speeding up the time it took to do some of the long calculations.