Michael Anilonis
Senior Capstone Project
CSC 482
Final Report

I. Introduction

A. Background

Machine learning is currently the newest technology taking the software world by storm. The possible capabilities of this new technology seems to be endless. I believe that finding new ways to implement this technology can ultimately be beneficial for our society.

In recent years, many countries have experienced revolutions and economic collapses. It is not crazy to hypothesize that the two may be related to one another. I stumbled upon the CIA World Factbook. The Factbook holds all of the declassified information about every country that the CIA has. I noticed right away that the majority of the data was economic or economically related. I decided this would be a great source of data for my machine learning models.

B. Purpose

With this CIA data, I decided that taking the economic and economically related data and putting it into a neural net could lead to some promising results. However, I knew that there had to something for this neural net to predict. I designed the neural net to predict the amount of years the country will exist given the information. This could be beneficial for any economist or historian looking for trends with how economics affects the stability of a country. My idea when creating these neural nets was to see if poor economics actually correlates with a country that falls within a certain number of years.

II. Development

A. HTML Parser

The first part of this project was to create a parser for the data I wanted to use. All of the public CIA Factbook was all encoded into HTML webpages. To retrieve this data I knew that I needed to parse these HTML pages. I had some previous experience in parsing webpages using the Java programming language. I decided that using the jsoup library would be my best option for ease of parsing. I was able to download all of the Factbook data via the CIA official website. I then created a general parser that was able to grab all the data and put it into a hash map. This Hash Map was then serialized onto disk so the program does not need to parse the data every time. In the end, all the data from 2000 to 2014 was parsed and can be accessed via the serialized hash map.

To see how this phase of the project was completed please visit: https://github.com/manilonis/Senior-Capstone-Project/tree/java-html-parser

B. RESTful API

Once I had the data serialized onto disk. I knew that the machine learning model had to access this data in some way. I also wanted any machine that was using the model to be able to access the data. I decided that the best way to do this was to create an API using REST protocol. I also chose to use java for this since it could then access the serialized data from the parser. I found that the Spark library for Java would be my best option to create a RESTful API in Java.

The decided to make all my endpoints hosted on pi.cs.oswego.edu. This server does not get a crazy amount of traffic and was easy for me to manage. The first endpoint I wanted was for the countries to be listed for every year. Thus, if the API is passed a GET request with just a year, the return JSON is a list of the countries that are available for that year.

The second endpoint is for the specific data for a specific country for a specific year. Thus, if you pass a GET request that has both a year and country, the exact data from the Factbook for that country for that year will be in the return JSON.

With these two endpoints, a user is able to gather all the possible data from the CIA Factbook RESTful API that I created.

To see how this phase of the project was completed please visit:

https://github.com/manilonis/Senior-Capstone-Project/tree/api-backend

C. Machine Learning Model

In my latest internship experience I was able to work with the Tesnorflow library in Python3. Therefore, I thought that would be the best course of action for this portion of the project. First I created an algorithm that sent GET requests to the API that I had created. I then took all the data and put it into python list and dictionaries so it was easy to access.

I knew that I had to decide on what data parameters would go into the neural nets. I knew inputting all the information into the neural net was not an option. Thus I decided arbitrarily that the best inputs would be average military budget, average employment rate, average GDP growth rate, average labor force, average population, average import to export ratio, average budget, and average external debt.

Eight models were then created by randomly assigning four data attributes to each neural net to train on. These models were then trained using a supervised learning algorithm. This is due to the fact that the results were highly skewed to the maximum number of years due to the fact that many countries never fall. Thus, if the neural net did not predict anything but a certain year, I would give it different weights to try and correct the learning of the neural network.

This method ended up working out very well. Where the accuracy of each model never reached below 86% even as I kept messing with the weights of the model during training.

D. Graphical User Interface

It is only beneficial to develop software if one can end up using said software. I decided that developing a graphical user interface would better allow any user to use my software. I developed the interface using the Python tkinter library. This was mainly because it is a very simple library and using Python would allow me to interface easier with the neural networks that I had created.

The main menu is used to select whether the user just wants to use the API or actually have the models predict something.

If a user selects the “Try the API” button, the user is shown a screen where they can select the year and country the user would like to look into. Then, all the data keys show up and the user can pick which data point they would like to view. The resulting data then appears on the screen.

If a user selects the “Predict Years of Existence”, they are brought to a window that will allow the user to use the neural networks that have been trained. The possible data attribute are shown with blank text boxes. The user can then enter the attributes they would like to try and predict. If the correct attributes are chosen and the correct type of data is inputted, the result of the prediction will be shown after the “Predict!” button is pressed. If anything goes wrong “Error” will display on the screen.

III. User Testing

User testing was done with approval from the SUNY Oswego Human Subjects Committee. Subjects were presented with the graphical user interface and asked to use all the features displayed. The subjects were then asked to fill out a google form that asked different questions about the interface.

Here are the results of those tests:

On a scale of 1-10, how easy was the interface to navigate?

7 /10: 40 %

8/10: 40%

10/10: 20%

On a scale of 1-10, how would you grade the overall design of the interface?

8/10: 40%

9/10: 20%

10/10: 40%

On a scale of 1-10, how easy was the interface to use?

8/10: 20%

9/10: 20%

10/10: 60%

On a scale of 1-10, how would you grade the overall appearance of the interface?

7/10: 20%

9/10: 40%

10/10: 40%

On a scale of 1-10, do you feel that you were able to get all the desired features out of this interface?

7/10: 40%

8/10: 20%

9/10: 20%

10/10: 20%

Overall, the human testing showed that the graphical user interface had accomplished it was supposed to do. Many subjects found the interface very usable and were able to use the program appropriately. This goes to show that the graphical user interface was a success.

IV. Conclusion

This project as a whole can be looked on as a success. The biggest measure for accomplishment of this project should be the statistics of the machine learning models. The models have had great accuracy when predicting how long a country will last. All of the models had accuracy of at least 80% for every model. This is known to be great accuracy for machine learning models. I was very pleased with these results. I believe that if these models were allowed to train even more, the predictions can even be better for the models if they are allowed to train for a longer length of time. The human subject testing has also shown that the graphical user interface was a success. Overall I can look at this project as a success and could potentially grow into something great with more training and more work on the machine learning side of the project.