Computer Science Capstone

Workouts with Machine Learning

Ryan Barnes
Student ID #000970605

Table of contents

Table of contents **2**

Section A **4**

Letter of Transmittal 4

Project Recommendation 6

Problem Summary 6

Application Benefits 6

Application Description 6

Data Description 7

Objective and Hypotheses 7

Methodology 7

CRISP-DM Phases 7

Funding Requirments 8

Stakeholder Impact 8

Data Precautions 8

Developer Expertise 9

Section B **9**

Problem Statement 9

Customer Summary 9

Existing System Analysis 9

Data 10

Project Methodology 10

CRISP-DM Phases 10

Project Outcomes 11

Implementation Plan 11

Evaluation Plan 11

Resources and Costs 11

Programming Environment 12

Environment Costs 12

Human Resource Requirements 12

Timeline and Milestones 13

Section C **14**

Section D **15**

Project Purpose 15

Datasets 15

Data Product Code 16
Hypothesis verification 17
Effective Visualizations and Reporting 17
Accuracy analysis 18
Application Testing 19
Application Files 19
Installation Guide 19
User’s Guide 20
Summation of Learning Experience 20

Section A

Letter of Transmittal

01 August 2022

Real Persons
Weights are Heavy Inc.
404 Real Rd
Boulder, CO 00000

Executive Persons,

The customer base of Weights are Heavy Inc. is steadily declining as more workout applications with additional functionality enter the market. Our application can no longer compete by only tracking our users' workout statistics. We are in desperate need of an overhaul that will bring us to the cutting edge of what technology has to offer. Our proposed project will address these issues by creating an automatic workout generator that provides the personal trainer experience to the user. The workouts will be created from the users' past data through our machine learning model that improves with every use.

This overhaul will provide an easy-to-use experience for the casual gym goer to the professional athlete. Through our app, the user will be able to walk straight into the gym and start exercising without any planning or research. This ease of access will open the market back open for our company assuring more users.

The proposed project will take an estimated 4 months to be completed costing $198,162 with a yearly maintenance cost of $10,000. Employing 7 specialists to create a top-of-the-line application.

Included with this letter is our detailed project proposal breaking down every aspect of the project. As well as a prototype for a single exercise that predicts the amount of weight a user can lift.

I personally will oversee the completion of this project. I will be utilizing my academic knowledge accumulated from the Bachelor of Computer Science degree program at Westerns Governors University. The coursework and projects have provided me with experience with databases, programming, and machine learning all of which are required to complete the proposed project. To organize and lead this project, I acquired the project management certifications Project+ from CompTIA and ITIL foundation from Axelos. In the addition to the leadership skills and the experience I gained from being a team leader in the US army, this project will be completed as advertised.

Looking forward to working with you,

Ryan J Barnes

Ryan J Barnes

Project Recommendation

Problem Summary

Our current application is only used to track workouts for a niche customer base. The lack of customer interest is due to the application only tracking personal data without utilizing the data in any useful manner. With the rise of other competitors with more functionality the amount of our concurrent users is decreasing. We propose overhauling our application by adding automatic workout recommendations, powered by machine learning. By developing our app to turn raw data into information, we will transform our business model.

Application Benefits

Our application is the perfect foundation for implementing machine learning to predict workouts for our users. We already store and track the users' progress and can quickly enhance the application with the proposed new features. Providing workout recommendations will integrate our app into each one of our customers' workout sessions by improving user engagement and functionality. With the addition of machine learning, we expect to boost the rate users will increase muscle mass. We believe this proposed overhaul will vastly expand our customer base and provide exceptional returns to our shareholders.

Application Description

Our proposed project would allow the user to walk straight into the gym and work out without any planning or know-how. The app will automatically create a tailored plan for users based on their past lifts and goals. Each exercise will have recommendations for the amount of weight to be lifted by the user. The user will be able to adjust a difficulty slider on the fly and still be able to enter their own data. Every workout, exercise, and repetition will improve the accuracy of the machine learning model.

Data Description

The data will be created by our customers for every workout they do at the gym through the application on their mobile devices. The data will be transferred to our SQL database from the customer's mobile device. We then can use these statistics to train our machine learning model to recommend the exercises and amount of weight to lift. Since individual capabilities can be quite different, the model will primarily use only the user's data to make predictions. The user will be asked to collect baseline statistics from a few exercises to help the model make its initial predictions. As more data is collected the model will become more accurate for that individual.

Objective and Hypotheses

We hypothesize that the neural network model will meet the business requirements of the project. The model must be able to recommend exercises and the amount of weight to be lifted using collected data with at least 90% accuracy. The algorithm will be evaluated during the development process discussed in Section B Evaluation Plan. These conditions must be satisfied or another machine learning model will be selected.

Methodology

The proposed project will be conducted using the CRISP-DM Agile methodology. Many of the project's features must be finished in sequential order. For example, the SQL database must be complete before predictions can begin. Using CRISP-DM Agile methodology, the project will be designed to be constructed in chunks using the vertical slicing technique. The small team will require the flexibility, granted by CRISP-DM Agile, to move between phases as the project advances.

CRISP-DM Phases

Business understanding

- Complete project plan

Data understanding

- Transfer the data from the old application into new database design

Data preparation

- Organize and normalize necessary data

Modeling

- Train and test the selected neural network model

Evaluation

- Assess if the model and data set to meet the requirements of the project

Deployment

- Complete documentation and prepare for live release

Funding Requirments

An estimated $198,162 is required to complete the proposed project and $10,000 a year for server hosting. The majority of these costs are for the 7 personnel that are required to develop the application over 4 months.

Stakeholder Impact

With the new features of the application will increase the customer base and charge price adding value to the shareholders. The self-building workouts will be convenient to the casual customer who normally does not use a workout tracker. With the addition of machine learning to the initial app, store costs can be increased or a subscription service implemented to reduce the new server maintenance costs. This project is the next step in the expansion of our company.

Data Precautions

The data will be stored locally on the user devices with the ability to opt into sharing some data to improve the machine learning model. The data used to train the model would include the users' statistics such as weight, height, age, etc., and the results of the exercise they did. To protect the users' privacy and confidentiality a random user ID would be used to organize the information.

Developer Expertise

To spearhead this project I will be utilizing my academic knowledge accumulated from the Bachelor of Computer Science degree program at Westerns Governors University. The coursework and projects have provided me with experience with databases, programming, and machine learning all of which are required to complete the proposed project. To organize and lead this project, I acquired the project management certifications Project+ from CompTIA and ITIL foundation from Axelos. In the addition to the leadership skills and the experience I gained from being a team leader in the US army, this project will be completed as advertised.

Section B

Problem Statement

We’ve found there is a common problem with all data science: what do we do with all the data we have collected? We propose overhauling our application by adding automatic workout recommendations powered by a neural network. Developing our app to turn raw data into information will finally put the data we have stored to profitable use.

Customer Summary

Our current application is only used to track workouts for a niche customer base on their mobile devices. The user must have outside knowledge to create their own workout routines. The proposed overhaul will remove know-how requirements and appeal to the wider market by automatically creating the routines. This will allow anyone, from casual gym goers to professional athletes, to use our application with ease. We will construct these recommendations by training a machine learning model with our existing customer data. Our hope is for our application to perform better than a personal trainer.

Existing System Analysis

With our existing application, the user data is simply stored without any processing functions. The user's stats are stored in a file on their mobile device created by the application. The application has simple create, update and delete functions. There are no unique identifiers per user so that will have to be added when migrating to our new database along with other alterations of the data structure.

Data

The organization of our SQL database will be in 3rd normalized form reducing data duplication and improving data integrity. Once the database design phase is complete is it imperative that we test it with the machine learning side of our project before large-scale data migration. Core database changes to align with the data processing could lead to costly setbacks if the database is filled too early. The application GUI will use choice boxes and sliders when possible to prevent unusual or incorrect entries from being made.

Project Methodology

The proposed project will be conducted using the CRISP-DM Agile methodology. Many of the project's features must be finished in sequential order, for example, the bounding box must be complete before identifying can begin. Using CRISP-DM Agile methodology, the project will be designed to be constructed in chunks using the vertical slicing technique. The small team will require the flexibility, granted by CRISP-DM Agile, to move between phases as the project advances.

CRISP-DM Phases

Business understanding

- Complete project plan

Data understanding

- Migrate data from the old application into new database design

Data preparation

- Trim and label collected footage

Modeling

- Train and test the selected neural network model

Evaluation

- Assess if the model and data set to meet the requirements of the project

Deployment

- Complete documentation and prepare for live release

Project Outcomes

With the completion of our proposed project the will be a ready-to-deploy application for our customers to use. The application will include all previous app functionality with the new ability to recommend workout routines created using machine learning. The application will include tutorials for old and new users on the new features. The database will be tested and ready for 100,000 users with the ability to scale as needed. During the development process, an installation and user guide will be provided for evaluators to test the application’s progress.

Implementation Plan

The project will be completed in 4 general phases. Phase 1 will consist of creating and testing the database and machine learning program. Phase 2 will be migrating the data from the old application into the new database. Phase 3 includes building the new GUI that interfaces with all the new features of the backend. The last phase will be complete unit testing and final documentation. After all these phases are completed the app will be able to go live and officially replace the old one.

Evaluation Plan

To ensure this project's development not only meets the requirements but surpasses the expected machine learning accuracy of 90%. A Quality Assurance Specialist will be part of the whole development process. Ensuring that throughout all phases every section is thoroughly tested. Phase 1 testing will ensure that the database and program work together to maintain a high level of data integrity. Phase 2 testing must formulate responses to any unusual or incomplete entries when migrating the old data to the new database. Phase 3 testing will the user frontend functionality and exploitative techniques are not possible for example SQL injections. Phase 4 testing will test the whole system and measure how it performs with simulated concurrent users. With testing built into every phase of our development, we will build a secure and easy-to-use application worthy of customer loyalty.

Resources and Costs

Below is an itemized table of the project's estimated costs.

Resource	Description	Cost
Developers	4 programmers (~80,000 yearly salaries at an estimated 600 work hours each)	$102,400
Database Administrator	1 Database builder (~90,000 yearly salaries at an estimated 600 hours)	$25,962
Quality Assurance	1 QA (~70,000 yearly salaries at an estimated 600 work hours)	$21,000
Senior developer	1 manager (~110,000 yearly salaries at an estimated 600 work hours)	$31,800

Database hosting	able to handle 100,000 users with 1,000 concurrent costs ~10,000 per year	$10,000
Software Budget	Budget for any software licensing that may be required	$1,000
	Total	$198,162

Programming Environment

The following software will be used during the development of this project. The software used has no additional cost but $1,000 has been set aside for licensing fees if required.

- Programing language: Python 3.10.1, C#
- Database: SQL Server 2019
- Version control: GIT 2.34.1
- Management: Microsoft Project
- OS: Windows 10
- IDE: Microsoft Visual Studios, JupyterLab

Environment Costs

The SQL server will be hosted by a reputable third-party data center at an estimated cost of $10,000 a year. It has been deemed more cost-effective to rent than to hire staff to construct and maintain a server and infrastructure at a physical location.

Human Resource Requirements

We estimate that the project will require 4 months to be fully completed with a well-rounded team of 7 engineers. The cost of labor will be the bulk of the budget for the project.

Timeline and Milestones

Milestones	Activity	Start	End
01	The proposal is accepted	01-Sept-22	-
02	Finalize project plan	01-Sept-22	10-Sept-22
03	**Phase 1 - Database and Program Construction**	10-Sept-22	20-Oct-22
04	Database design	10-Sept-22	20-Sept-22
05	Database purchase and setup	20-Sept-22	25-Sept-22
06	Framework for machine learning program	20-Sept-22	30-Sept-22
07	Predictions are made using test data from database	30-Sept-22	15-Oct-22
08	Unit testing	15-Oct-22	20-Oct-22
09	**Phase 2 - Data Processing and Migration**	20-Oct-22	15-Nov-22
10	Create a process to prepare old data for new database	20-Oct-22	30-Oct-22
11	Unit testing	30-Oct-22	10-Nov-22
12	Transfer data into database	10-Nov-22	15-Nov-22
13	**Phase 3 - Frontend Creation**	15-Nov-22	10-Dec-22
14	Update user GUI to modern standards	15-Nov-22	05-Dec-22
15	GUI includes new ML predictions	27-Nov-22	05-Dec-22
16	Unit testing	05-Dec-22	10-Dec-22
17	**Phase 4 - Final Testing and Documentation**	10-Dec-22	20-Dec-22
18	Complete application testing	10-Dec-22	20-Dec-22
19	Review all documentation of database, programming, and project	10-Dec-22	20-Dec-22
20	**Project Completion**	20-Dec-22	-

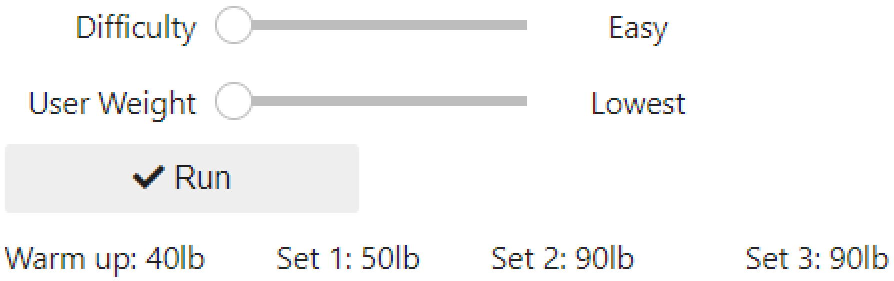
Section C

The program files are attached separately to this document.

Section D

Project Purpose

The purpose of this project is to address the absence of data mining functions in the client's workout application. Machine learning was utilized to turn the data into information by predicting the amount of weight a user should lift based on past workouts. In the sections below, a prototype program is examined that is trained on a single user and exercise to prove the feasibility of the project. The user would adjust the sliders and be able to lift the recommended weight per set. Below is an example of how the user GUI functions.



Difficulty ○ Easy

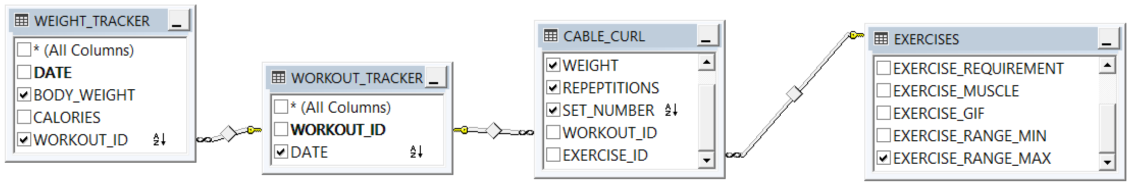
User Weight ○ Lowest

✓ Run

Warm up: 40lb Set 1: 50lb Set 2: 90lb Set 3: 90lb

Datasets

The data is currently stored within a SQL database, concentrated into a view, and exported into a CSV file for ease of access to the test build. The CSV is included in the root with the JuypterLab file titled curling_dataset.csv. The data set used in this example is small and contains less than 100 entries. The data is stored on the local device and will not run the risk of interception that comes with the transmission. Based on testing normalizing the data did not increase the accuracy of the model and it was decided to leave the data as is. The data below is used to make predictions on the weight column.

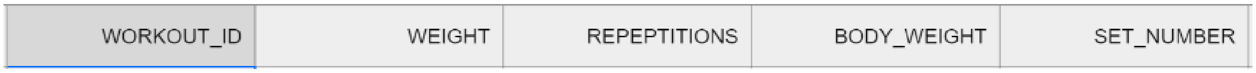


WEIGHT_TRACKER
- □ * (All Columns)
- □ **DATE**
- ☑ BODY_WEIGHT
- □ CALORIES
- ☑ WORKOUT_ID

WORKOUT_TRACKER
- □ * (All Columns)
- □ **WORKOUT_ID**
- ☑ DATE

CABLE_CURL
- ☑ WEIGHT
- ☑ REPEPTIONS
- ☑ SET_NUMBER
- □ WORKOUT_ID
- □ EXERCISE_ID

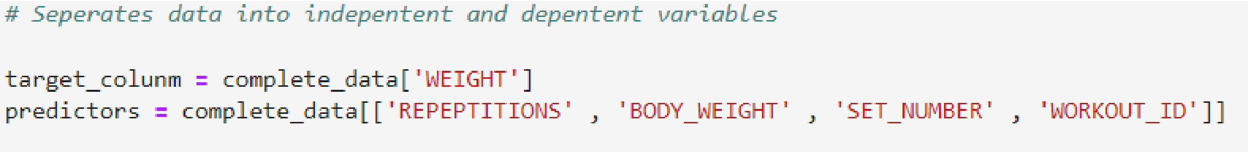
EXERCISES
- □ EXERCISE_REQUIREMENT
- □ EXERCISE_MUSCLE
- □ EXERCISE_GIF
- □ EXERCISE_RANGE_MIN
- ☑ EXERCISE_RANGE_MAX



WORKOUT_ID	WEIGHT	REPEPTITIONS	BODY_WEIGHT	SET_NUMBER

Data Product Code

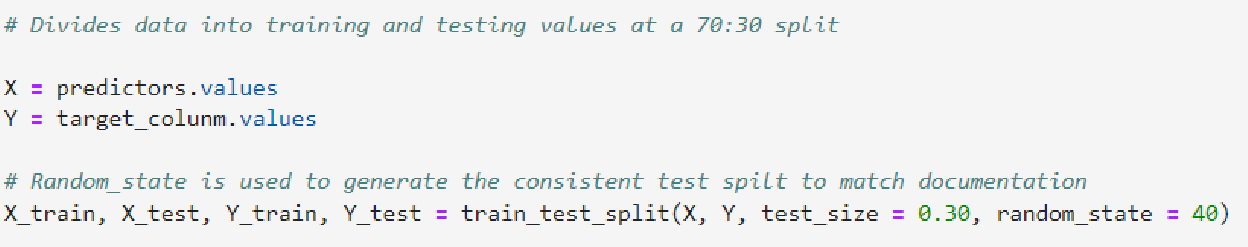
The data from the CSV file is broken into independent and dependent variables, the lift weight being the target variable for the ML. The 4 values used for the prediction are repetitions, user body weight, set number, and workout ID from past workouts. Normalizing the data was found to decrease accuracy during testing and the code for it is left in but is commented out.



```
# Seperates data into indepentent and depentent variables

target_colunm = complete_data['WEIGHT']
predictors = complete_data[['REPEPTITIONS' , 'BODY_WEIGHT' , 'SET_NUMBER' , 'WORKOUT_ID']]
```

The data is split into training and testing values at a ratio of 7:3 and the split remains persistent to match this documentation. The sklearn’s train_test_split function is used to achieve this.

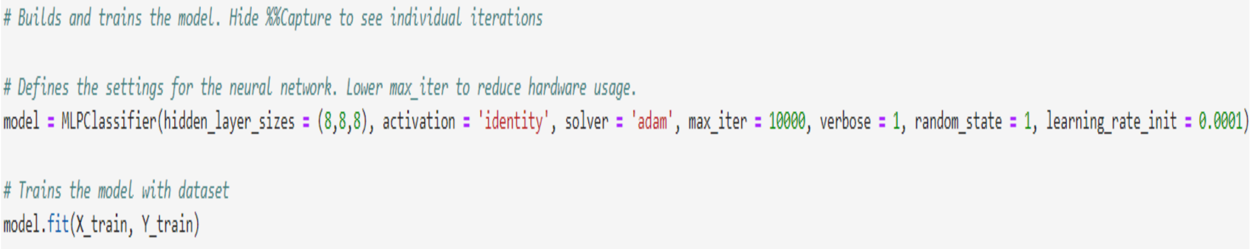


```
# Divides data into training and testing values at a 70:30 split

X = predictors.values
Y = target_colunm.values

# Random_state is used to generate the consistent test spilt to match documentation
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.30, random_state = 40)
```

The data is then used to train the neural network. The model is set to run for 10,000 iterations stopping if the minimum improvement value of 0.0001 is not met. Adam is an optimizer that combines Momentum and RMSprop Optimizers allowing the model to train faster.

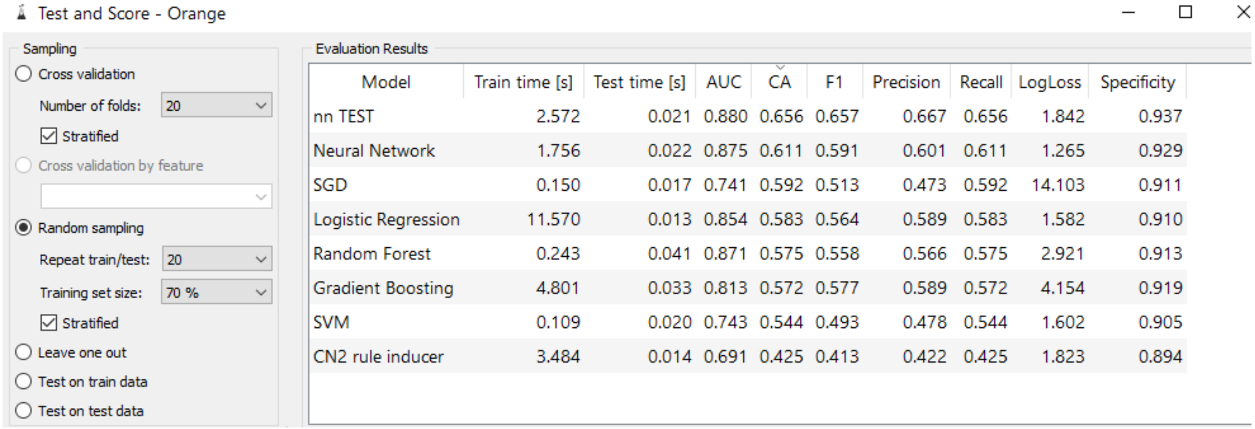


```
# Builds and trains the model. Hide %%Capture to see individual iterations

# Defines the settings for the neural network. Lower max_iter to reduce hardware usage.
model = MLPClassifier(hidden_layer_sizes = (8,8,8), activation = 'identity', solver = 'adam', max_iter = 10000, verbose = 1, random_state = 1, learning_rate_init = 0.0001)

# Trains the model with dataset
model.fit(X_train, Y_train)
```

The neural network algorithm was selected for the project through testing in the program Orange 3 data mining. Orange 3 enables rapid testing without the need for coding through their GUI. With the highest CA(classification accuracy) the neural network settings were then tweaked increasing the CA by 5% shown as nn test. Those options were then used in the program's model as seen above in the MLPClassifier settings.



Sampling

○ Cross validation

Number of folds: 20

☑ Stratified

○ Cross validation by feature

● Random sampling

Repeat train/test: 20

Training set size: 70 %

☑ Stratified

○ Leave one out

○ Test on train data

○ Test on test data

Evaluation Results

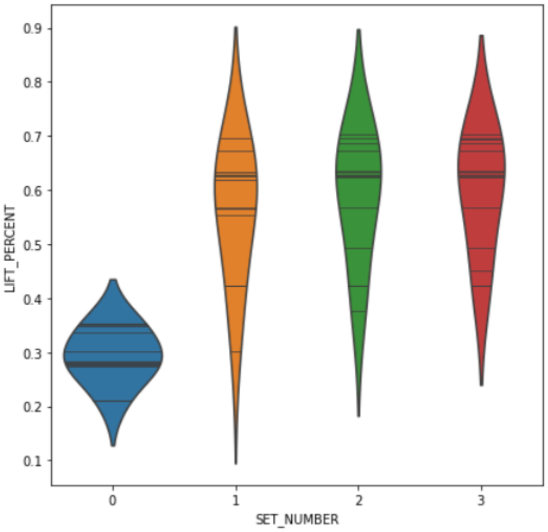
Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
nn TEST	2.572	0.021	0.880	0.656	0.657	0.667	0.656	1.842	0.937
Neural Network	1.756	0.022	0.875	0.611	0.591	0.601	0.611	1.265	0.929
SGD	0.150	0.017	0.741	0.592	0.513	0.473	0.592	14.103	0.911
Logistic Regression	11.570	0.013	0.854	0.583	0.564	0.589	0.583	1.582	0.910
Random Forest	0.243	0.041	0.871	0.575	0.558	0.566	0.575	2.921	0.913
Gradient Boosting	4.801	0.033	0.813	0.572	0.577	0.589	0.572	4.154	0.919
SVM	0.109	0.020	0.743	0.544	0.493	0.478	0.544	1.602	0.905
CN2 rule inducer	3.484	0.014	0.691	0.425	0.413	0.422	0.425	1.823	0.894

Hypothesis verification

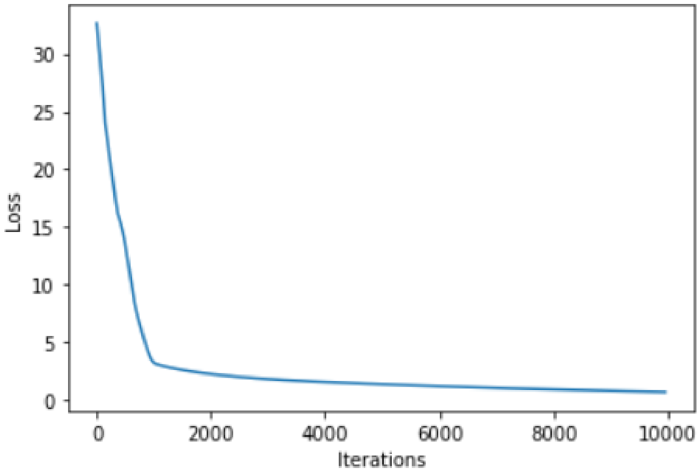
The finished project goal of 90% accuracy and above was not yet achieved. This outcome was expected due to the small dataset used to train the model. However, even with the limited data, the accuracy of the model reached around 80%. The program was able to meet all the goals proposed in the topic approval form.

Effective Visualizations and Reporting

To help visualize the data and analytical process several graphics are created during the execution of the program. The disruption of the dataset is described using a violin plot. The violin plot visualizes the differences in each lifting set the model must account for as the user gets tired or increases weight. A key indicator for the model was identifying that set 0 or the warm-up was less than the rest of the sets.

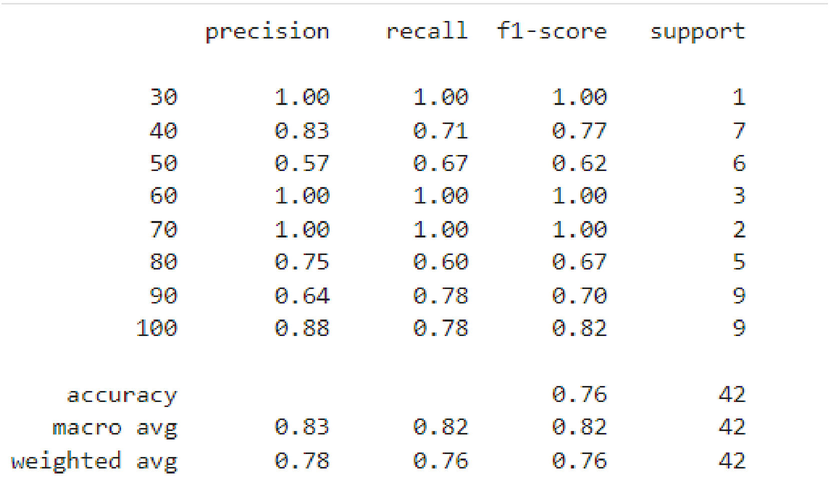


To visualize the non-descriptive method of the model being trained a loss curve graph is used. This graph is also a key indicator of the model's health. The loss curve decreases as it approaches its limit of 0 describes the model continuing to improve. If the model wasn’t improving the graph would be sporadic with a zig-zag pattern with its predictions randomly being made with no improvement.



Accuracy analysis

As the classification report shows below the model prediction accuracy is around 80% which is good for such a limited dataset. The GUI sliders produce realistic numbers when changed for weights for the exercise. However, the model does not increase the weight past previous max values even when the slider's difficulty and weight are maxed out. The issue must be addressed in future versions of the program so the lifter can be continually challenged.


```
              precision    recall  f1-score   support

          30       1.00      1.00      1.00         1
          40       0.83      0.71      0.77         7
          50       0.57      0.67      0.62         6
          60       1.00      1.00      1.00         3
          70       1.00      1.00      1.00         2
          80       0.75      0.60      0.67         5
          90       0.64      0.78      0.70         9
         100       0.88      0.78      0.82         9

    accuracy                           0.76        42
   macro avg       0.83      0.82      0.82        42
weighted avg       0.78      0.76      0.76        42
```

Application Testing

I broke the project into three sections building and testing the input, processing, and output. Each section required the prior section to function and was tested before moving to the next part. To improve the accuracy of the predictions I tested a variety of settings that controlled how the neural network model was created. Even testing how the data's organization could affect the prediction. Normalizing the data into a range between 1 and 0 can increase accuracy for example but through testing, I found normalizing caused a 5% loss in accuracy in my model.

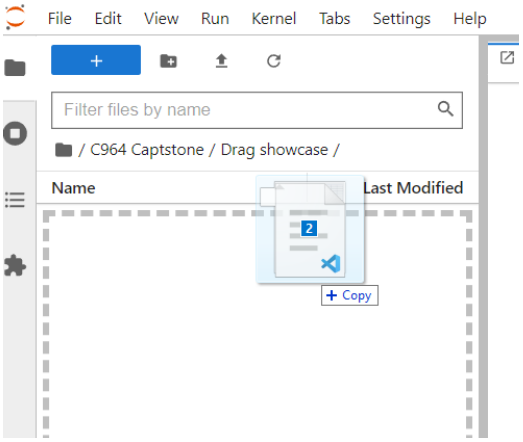
Application Files

2 files are included in the project folder

\weight_predictor.ipynb
\curling_dataset.csv

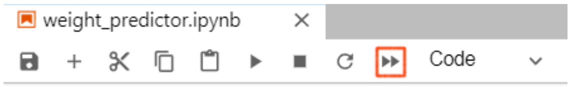
Installation Guide

1. Install Anaconda Distribution
2. Install Jupyter Lab from the Anaconda application
3. Download the included zip with the project submission
4. Unzip and drag the two files, weight_predictor.ipynb and curling_dataset.csv, into Jupyter Lab

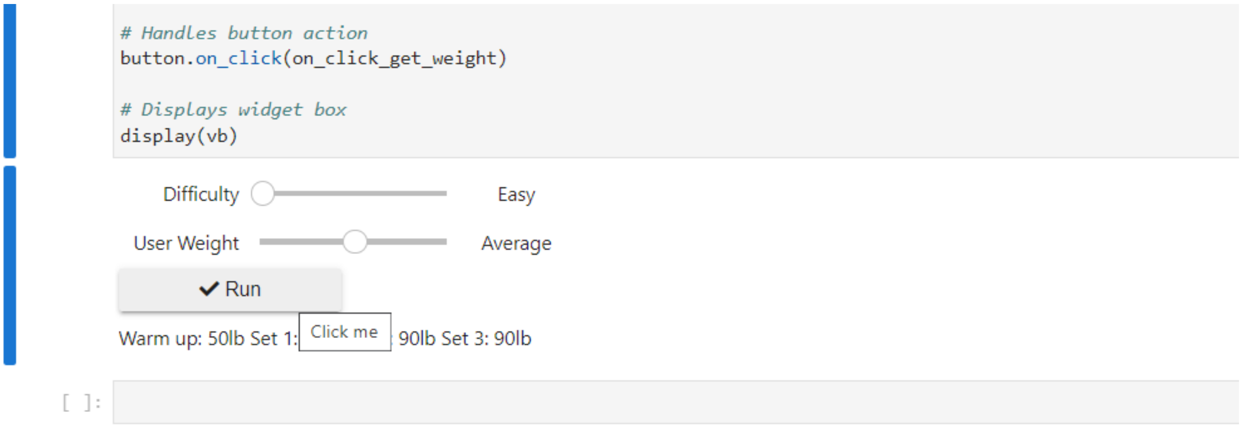


User's Guide

1. Double click weight_predictor.ipynb to open the Notebook in Jupyter Lab
2. Press the double play button to restart Kernel and run all cells
3. Wait for the libraries to be installed and the code to execute
4. The user interaction is located at bottom of the notebook with the option sliders



5. Once the difficulty and weight slider are set, click the button right beneath the slider to view the model's predictions



```
# Handles button action
button.on_click(on_click_get_weight)

# Displays widget box
display(vb)
```

```
Warm up: 50lb Set 1: 90lb Set 3: 90lb
```

Summation of Learning Experience

This project was my first experience with machine learning and graphics in python taking quite a bit of research. The first step was handling the dataset. I built a database in Microsoft SQL Server Manager to store the data and created a view with all the required information. Fortunately, there have been several courses in SQL for my degree plan providing the knowledge for me to focus on database design and data entry.

Initially, I was using a program called Orange 3 to manipulate the data and algorithms to get a better understanding of what I needed to do. Orange 3 allowed me to quickly test which graphs and algorithms worked best for my project without doing any coding. Then came the research on how to present the code, resulting in my choice of Jupyter Lab.

Ultimately this project required me to research and identify the right tools for the job which is an important skill in itself. I plan to continue the development of this project and hope to reach the functionality described in sections A and B. This project even before completion has helped me get a job a as Web Developer by providing talking points and real-world skills.