

C964: Computer Science Capstone

Task 2 parts A, B, C, and D

Daniel Reeve

001990892

Part A: Project Proposal for Business Executives	3
Letter of Transmittal	3
Project Recommendation	4
Problem Summary	4
Application Benefits	4
Application Description	4
Data Description	4
Objectives and Hypothesis	5
Methodology	5
Funding Requirements	6
Data Precautions	6
Developer's Expertise	6
Part B: Project Proposal	7
Problem Statement	7
Customer Summary	7
Existing System Analysis	7
Data	7
Project Methodology	8
Project Outcomes	9
Implementation Plan	9
Evaluation Plan	9
Resources and Costs	9
Timeline and Milestones	11
Part C: Application	13
Part D: Post-implementation Report	14

A Business (or Organization) Vision	14
Datasets	14
Data Product Code	15
Objective (or Hypothesis) Verification	16
Effective Visualization and Reporting	16
Accuracy Analysis.....	16
Application Testing	16
Application Files.....	17
User Guide	17
Summation of Learning Experience	18
Sources	19

Part A: Project Proposal for Business Executives

Letter of Transmittal

March 7, 2023

Daniel Reeve, Lead Software Developer
Employment Inc.
555 Applewood Ln
Duncansville, Pennsylvania

Dear Mr. Michaels,

As increased numbers of people obtain college educations, those individuals will be seeking job opportunities. Therefore, businesses are looking to simplify the process of evaluating a candidate's likelihood of employment. Employment Inc. is one of those businesses. Our hiring staff are having a challenging time processing through each application because they require so much manual evaluation. A solution needs to be found to help our staff manage their time and performance in a better manner.

Therefore, I recommend the following solution. Using machine learning and having a user inputting specific data fields, such as the test score percentages we ask for on each application, Employment Inc. will be able to predict if an individual is likely to receive a job or not in a new application to be developed. We have found that there is a correlation in test scores and whether an individual is likely to get a job or not. Therefore, the application will take the test scores the user inputs and predict their employment likelihood. This will provide several benefits to the company.

The solution will benefit the company in a few diverse ways. The hiring staff will be able to process through applications at a faster rate because the application will be able to eliminate some of their manual processes currently being employed within the company. A more intelligent staff will also be created as a result of the application because people with better test score averages will be those that tend to get predicted as employable.

The estimated cost of the proposed software solution is around \$21,500. This includes the costs of software, hardware and labor required to develop and implement the application.

I would be a qualified individual to develop this solution because I have already conducted the research needed for development. I have also been the lead software developer for the company for 5 years and have developed and implemented many new projects that have been beneficial to the company. My education consists of a Bachelor of Science in Computer Science, along with many certifications in software development applications. With my knowledge and experience in the field, I believe I would be the best fit developer to tackle this project.

Sincerely,

Daniel Reeve

Project Recommendation

Problem Summary

The issue that Employment Inc., among other businesses, is running into is a large influx of employment applications, but not enough processing resources to manage it. We ourselves are having a challenging time processing through applications to determine if an applicant is employable or not, but a software solution is the answer we seek. The solution I propose is a software application that predicts if an applicant is employable based on several of their test scores. This software application will be able take in the 5 test scores we ask all applicants to provide on their application and predict if they are employable. This will help to reduce the amount of manual labor and time our hiring staff needs for each application, which they so desperately need.

The solution will be delivered as a web-based application that can be accessed with any web browser. I will develop the entire application and implement it within the company. The prediction tool will take in the 5 test scores as input and output if that particular applicant is likely to be employed. With the use of the tool, the hiring staff will be able to save themselves time and they will be able to identify more intelligent applicants for hire, which in turn provides the company with a more knowledgeable staff.

Application Benefits

- Describe (in more detail than above) how the project meets the business's (or organization's) needs.
- Describe how the business (or organization) will benefit from implementing the proposed solution.

The software solution will meet the business's needs because it will process applicant data for the hiring staff. No longer will the hiring staff have to spend time on applications that do not meet a satisfactory level of test scores. We need the hiring staff to be able to focus on the applications of the candidates that are most likely to be appropriate for a job. With the knowledge the tool provides, the hiring staff will be able to focus on those applications rather than waste time on other candidates. This will allow them to process through the influx of applications at a faster rate.

A benefit to the hiring staff with the implementation of this software solution is that they will be saved a lot of time and effort. They are currently evaluating the test scores manually to determine if the candidate tested high enough for employment. The software application will do that processing for them, which saves the hiring staff time and energy. With the result the software application provides, the hiring staff will also be able to discard the applications that do not return a likelihood of employment. Thus, saving from further processing and interviewing of unqualified applicants. The company will also benefit from the software solution because more intelligent staff will be hired. More intelligent staff can result in better products and services, which can increase profits.

Application Description

- Provide technical details on how the application will solve the problem.

Data Description

The dataset to be used will be taken from kaggle.com, a free web based and public dataset application. The dataset that we will be using is called "Job Placement Dataset" and it has several cases of

test scores we look for, along with the employment status of those cases. The data types we will use are as follows:

- ssc_percentage (float)
- hsc_percentage (float)
- degree_percentage (float)
- emp_test_percentage (float)
- mba_percent (float)
- status (object)

Independent variables:

- ssc_percentage
- hsc_percentage
- degree_percentage
- emp_test_percentage

Dependent variables:

- status

With the dataset, I do not see any outliers that would prevent the software application from performing its expected tasks. However, if there are any, they will be removed from the dataset so the results of the machine learning model will not be skewed. The dataset does have the limitation of being a smaller finite set of data and there are no dates to show when these cases were obtained. In the future we may mitigate that by saving all the applicant data the hiring staff enters into a database to further train the machine learning model.

Objectives and Hypothesis

The application will be able to predict the employability of an applicant with an 80% or better prediction rate. The application will be easy to use for the hiring staff and will simplify their applicant processing. It will also show visuals of the data being used from the dataset, so the hiring staff can get a better understanding of how the data is being used and what it represents. Through all of this, the ultimate objective is to provide a means of improvement for the business as a whole, through the facilitation of quicker and better hiring.

Methodology

The waterfall development methodology will be used to develop and implement this project. The waterfall method was chosen because the project requirements are well defined prior to the start of development and implementation. With very little, to no changes, the agile method is not needed for this project.

The waterfall method has the following common phases:

- Requirements:
 - The requirements phase is the period of time needed to acquire and understand certain aspects of the project. As it pertains to this project, we will be meeting with end users to discuss the requirements they are looking for and the outcomes needed from the proposed application.

- Design:
 - The design phase is when the projects scope and functionality are designed. We will use this phase to design an outline of the products design path and make sure that all end user requirements are met.
- Implementation:
 - The implementation phase is where the project is developed. We will be writing all the necessary code for the product in this phase, in accordance with the requirements previously gathered.
- Testing:
 - The testing phase is where all requirements are verified to be present and there are no errors in the application. We will be using a sample set of data to verify that the application produces the correct outputs. We will also present the application to a small group of end users for them to verify all requirements are met.
- Deployment/Maintenance:
 - In this phase, the product will be released, and any errors discovered will be fixed over time. This is when we will launch the application for use by the end users. Any errors or bugs they report will be fixed as needed.

Funding Requirements

The funding required for this proposal equates to an estimated sum of \$21,500. This includes all the hardware, software, and labor costs associated with the development and deployment of the application. Most of the software and programming languages are open source and free to use. Some hardware pieces, such as a laptop, are required for the developer. The labor costs consist of the salary of the developer for the duration of the project, as well as the internal end user's salary for testing.

Data Precautions

The data in the dataset does not show who the case is representative of beyond the gender of the individual, so it does not contain any sensitive or protected data. This is how Kaggle.com is able to provide the dataset as a free and public product. That being said, the gender will still not be used in our application, so it will be discarded prior to using the dataset. The dataset is public to use, so the even if we did use the gender field, we would not be using data that is freely visible to anyone in the world. Therefore, the application will still be in compliance with all privacy laws.

Developer's Expertise

I would be the best fit and most qualified developer for the development and implementation of this project because of my experience and education in the field. I have a Bachelor of Science in Computer Science, along with many certifications in software development applications. These certifications consist of PCPP2: Certified Professional in Python Programming 2, PCAT: Certified Associate in Testing with Python, and many others. These skills will help with the development of the software solution because the application will be built using Python as the primary programming language. I have also been with the company as the lead software developer for 5 years. In that time, I have designed, built, and implemented many custom software solutions within the company. One of which had a similar development process and method to the ones being used for this project.

Part B: Project Proposal

Problem Statement

The issue that Employment Inc., among other businesses, is running into is a large influx of employment applications, but we do not have enough processing resources to manage it. As a company, we are having a challenging time processing through applications to determine if an applicant is employable or not. We need a solution to free up time for our hiring staff and we need to hire more intelligent staff members.

Customer Summary

The clients affected by the proposed solution are our hiring staff, which consists of a team of four full time employees that do all the hiring for our company. They struggle to process through the applications we receive every day. Most of their processes are manually done, so a lot of time is spent trying to determine if an applicant is employable by the company. They would be the ones using the application on a daily basis.

The solution I propose is a software application that predicts if an applicant is employable based on several of their test scores. This software application will be able take in the five test scores we ask all applicants to provide on their application and predict if they are employable. This will help to reduce the amount of manual labor and time our hiring staff needs for each application. With the use of the proposed application, the hiring staff will be able to save themselves time and they will be able to identify more intelligent applicants for hire, which in turn provides the company with a more knowledgeable staff.

Existing System Analysis

The hiring staff manually evaluates the test scores they receive from each applicant, which takes a lot of time and energy. There is currently no technological solution in place for their processes because we have never had such a large influx of applications before. Their current methods were working, but with more and more applications coming in every day, they are falling behind in their work.

The proposed solution is needed to help them keep pace with the number of applications that are coming in. With all processing being done manually, a lot of time is wasted in a day. If some applications could be removed from the candidate pool based on the results of the proposed application produces, then the hiring staff would be able process the applications in a more time efficient manner. A lot of manual labor would be reduced, because not only will the process of manually evaluating test scores be simplified, but the applications that get rejected will not have to further processed.

Data

The dataset will be obtained from Kaggle.com, an open source and free dataset application. It will be downloaded in a csv format, so it will already be the format we need for the application to process it. However, some columns of data will need to be removed because they do not pertain to the machine

learning model we are going to develop. The only fields we would need to keep are those of the test scores and the employment status of each individual within the dataset.

How the data will be processed in each development phase:

- Requirements:
 - The data will not be processed or collected in this phase
- Design:
 - The data will not be processed or collected in this phase
- Implementation:
 - The data will be collected and processed in this phase
 - Before the application can process the data, some data fields will need to be removed from the file
- Testing:
 - A small sample of the dataset will be used for testing to ensure the application performs as required
- Deployment/Maintenance:
 - As maintenance occurs a small sample of the data set will be used to verify the application works properly

Any outliers or incomplete data will be removed from the dataset before being loaded into the application. However, looking at the dataset, there are no outliers or incomplete data that can be seen at first glance.

Project Methodology

The development methodology to be used for this project is the waterfall method. It incorporates five phases which are as follows: requirements, design, implementation, testing, and deployment/maintenance. All five phases will be utilized in the development of this proposed application.

The requirements phase is where project requirements are defined, such as costs, risks and timelines. Design is the phase that project scope and project mock-ups are created. The implementation phase is where the project is developed according to the requirements and design phases. Testing is then performed to make sure the project does not have errors or bugs. Then, the project is ready to be deployed for end user use in the deployment phase. Along with the deployment phase, general maintenance is performed.

How the phases will be utilized:

- Requirements:
 - Meet with end users
 - Discuss and gather the requirements needed
 - Discuss outcomes needed
- Design:
 - We will use this phase to design a mock-up of the product
 - We will also create a design path
 - Make sure that all end user requirements from previous phase are met
- Implementation:

- Prepare the dataset for use in the model
- Use design from previous stage to write all the necessary code for the product
- Make sure all requirements previously gathered are met
- Testing:
 - Meet with end users
 - Use a sample set of data to verify that the application produces the correct outputs
 - Have end users utilize the application to verify all requirements are met
- Deployment/Maintenance:
 - Released finished product to the end users
 - Continually collect and address errors or bugs that are reported

Project Outcomes

The deliverables at the end of this project consist of the finished application, to include the dataset used in both the original and altered format, and a user guide. The finished application will be able to be run by any end user that needs to run it. The altered dataset is the dataset that will be used for training the machine learning model and needs to be able to be accessed when the application is running. A user guide will be provided to explain how a user can access and run the application.

Implementation Plan

The software solution will be implemented by distributing the user guide to each hiring staff employee for review a few days before the deployment of the application. The application will then be uploaded to the appropriate GitHub repository as a single commit. At the time of deployment, a meeting will be set between the developer and the hiring staff for training. We will also review how the application fits into their everyday workflow by replacing some of their manual processes. The only dependency for the deployment is that the application is finished and has been approved by the client (hiring staff).

Evaluation Plan

Evaluation the finished product for errors or bugs, as well as requirements being met, is an essential aspect of the development process. While developing the product, the machine learning model will be tested with a sample set of data to ensure it meets the 80% accuracy rating needed. Then, once the application is completed, testing will be performed to ensure there are no bugs or errors present before release. All components of the application will be cross referenced against the requirements list developed in the requirements phase of the project.

To evaluate the performance of the application in solving the problem presented, monthly internal reports will be pulled on each employee on the hiring team to see how many applications they were able to process though in a one-month time period. These reports will be cross referenced against reports pulled in the months before the deployment of the application. The metric to be evaluated is if they are able to process through applications faster and still hire quality employees.

Resources and Costs

Hardware and Software Costs

Resource	Description	Cost
Dataset	Data to be sourced from Kaggle.com	\$0
ThinkPad T15p Gen 3 Intel (15")	Need 1 new laptop for the developer to work on	\$2,000
Programming languages and libraries	Python and Python libraries	\$0
Jupyter Notebook	Where the application will be developed	\$0
GitHub Account	Used for version control throughout the life of the project	\$0
	Total	\$2,000

Labor Time and Costs

Resource	Description	Time	Cost
Developer and End Users	Planning Phase	15 hours	\$2,000
Developer	Design Phase	15 hours	\$1,125
Developer	Implementation Phase	75 hours	\$5,625
Developer	Testing Phase	10 hours	\$750
Totals		115 hours	\$9,500

Environment Costs

Resource	Description	Cost
Maintenance	General maintenance for the life of application	\$10,000/year
	Totals	\$10,000

Timeline and Milestones

The following is the projected timeline of the project:

Planned Schedule

Milestone	Start	End	Resources	Tasks
Requirement Acquisition	2/20/23	2/28/23	Developer and End Users	The requirements will be acquired from the end users
Application Mock-up	2/28/23	3/6/23	Developer	The application will be designed to fit all requirements
Application Built	3/6/23	3/31/23	Developer	The application will be built to fit all requirements
Application Testing	3/31/23	4/7/23	Developer and End Users	The application will be tested for errors or bugs
Application Deployment	4/7/23	4/8/23	Developer and End Users	The application will be deployed for use by the end users
Maintenance	4/8/23	Undeterminable	Developer	End date of general maintenance is underminable because maintenance will be

				provided as long as the application is in use by the company
--	--	--	--	---

Part C: Application

The application can be accessed through mybinder.org with the URL <https://github.com/dreev65/WGU-Capstone>. For more detailed instructions, please see the “User Guide” section in Part D.

The GitHub repository contains the following two files related to the built application:

- WGU Capstone.ipynb ← application itself
- job_placement_data.csv ← dataset used

Part D: Post-implementation Report

A Business (or Organization) Vision

Our hiring staff were having trouble processing through high quantities of applications of possible new hires. Therefore, they needed an application that could help simplify their processes by taking in an individual's data and determining if they would typically be hired when compared to others.

The developed application was able to solve this problem by employing a Support Vector Machines (SVM) algorithm creates a prediction for the user based on their input. The model was trained with a dataset of 215 cases, then tested with a size of 0.3. The SVM takes this knowledge and applies it to the user input of 5 specific test scores to predict if the test scores would place the possible employee in a job or not.

Within the application, there is an "User Interaction" section close to the bottom that will ask the user for input. The input that is asked of are the 5 test scores of the possible employee. The user would first have to obtain those test scores from the applicant, which is already on their job applications to begin with. Then, the user can go to the "User Interaction" section of the application, enter the 5 test scores as percentages separated by a single space. Then they must hit the enter key on their keyboard to display the "Placed" or "Not Placed" text below the input section. The user can then use that knowledge when processing through the applications to determine if an application needs closer personal review or if they can remove that application from the possible new hires. Thus, the process is simplified for the user as they have less manual processing and review to conduct.

Say an employee had three applications come in, all those applications would have had the required test scores on them. The employee took those test scores, entered them into the application, and determined that one of the applicants is not predicted to get a job. They then threw out that applicant's application. They effectively cut the amount of manual processing down by a third, which increases productivity.

Datasets

The raw data was downloaded as a csv file from kaggle.com. Within the dataset, it contained several fields that were not necessary for the purpose of the solution to the proposed problem. It contained several object fields were not important to the solution, such as gender and what board of education a test score came from, so they were removed from the dataset. Once those fields were removed from the csv file, the file was saved and added to the application project folder. From there, I was able to have the application read the csv file into a two-dimensional list upon execution using a Python script. Once in the list, the data was separated into two variables, one for the independent data fields and the other for the dependent data fields (prediction field). Those two variables were then input into the machine learning model for training.

Example of unprocessed data:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	gender	ssc_percentage	ssc_board	hsc_percentage	hsc_board	hsc_subject	degree_percentage	undergrad_degree	work_experience	emp_test_percentage	specialisation	mba_percent	status	
2	M	67	Others	91	Others	Commerce	58	Sci&Tech	No	55	Mkt&HR	58.8	Placed	
3	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	
4	M	65	Central	68	Central	Arts	64	Comm&Mgmt	No	75	Mkt&Fin	57.8	Placed	
5	M	56	Central	52	Central	Science	52	Sci&Tech	No	66	Mkt&HR	59.43	Not Placed	
6	M	85.8	Central	73.6	Central	Commerce	73.3	Comm&Mgmt	No	96.8	Mkt&Fin	55.5	Placed	
7	M	55	Others	49.8	Others	Science	67.25	Sci&Tech	Yes	55	Mkt&Fin	51.58	Not Placed	
8	F	46	Others	49.2	Others	Commerce	79	Comm&Mgmt	No	74.28	Mkt&Fin	53.29	Not Placed	
9	M	82	Central	64	Central	Science	66	Sci&Tech	Yes	67	Mkt&Fin	62.14	Placed	

Example of processed data:

	A	B	C	D	E	F	G
1	ssc_percentage	hsc_percentage	degree_percentage	emp_test_percentage	mba_percent	status	
2	67	91	58	55	58.8	Placed	
3	79.33	78.33	77.48	86.5	66.28	Placed	
4	65	68	64	75	57.8	Placed	
5	56	52	52	66	59.43	Not Placed	
6	85.8	73.6	73.3	96.8	55.5	Placed	
7	55	49.8	67.25	55	51.58	Not Placed	
8	46	49.2	79	74.28	53.29	Not Placed	

Both the processed and unprocessed data files can be accessed by going to <https://github.com/dreev65/WGU-Capstone> in any browser, then clicking the file you would like to view. To see the original download location of the dataset, you would have to go to the following URL in any browser: <https://www.kaggle.com/datasets/ahsan81/job-placement-dataset>

Data Product Code

The application does not process the raw data on its own. The dataset was manually altered before the application got to process it. This was done by removing unnecessary object fields to ensure that only the relevant fields were being used in the application. Once the application got the file, it read it and stored each row in a two-dimensional list for use in the model. The data is stored in a two-dimensional list because it makes it easier to separate the prediction from the model data when processing.

The descriptive methods used consist of histograms for each of the test scores in the dataset, a scatter plot matrix, and a pie chart. The histograms show the distribution of the test score percentages along the x-axis and the number of people along the y-axis. The scatter plot matrix shows the histogram of each set of test scores, as well as the scatter plot for each pair of test scores. This was useful in developing the model because it shows a correlation between two tests, and it can be determined what test scores are important within the model. The pie chart shows the percentage of individuals in the data set that were placed in a job and those not placed. This helped with the development processes because it showed a representation of the data used and gave better insight on what results to expect when testing the model.

The non-descriptive method used for the project was a prediction of new applicant likelihood of being placed or not placed in a job based on the testing scores they provided. How this was performed was by having the user obtain the 5 required test scores from the applicant, then entering them into the application as input. That input passed to the prediction method of the application, where the machine learning model would determine the prediction based on the knowledge it has from training and testing. The result of the prediction would then be printed out for the user to see. To verify that the prediction was

accurate enough for the solution, three separate models were trained and tested. The Support Vector Machines (SVM) algorithm was the most accurate out of the three models tested, so that is the model that was used in the finished application. All the models were trained and tested with the same dataset and the accuracy was evaluated by an accuracy score metric. The reason the prediction was chosen as the non-descriptive method is because the dataset was able to show a correlation between the test scores and employment status. Based on an individual's test scores, the likelihood of that individual getting or not getting a job could be inferred by the model, using the knowledge it has of others' employment status and test scores.

Objective (or Hypothesis) Verification

The objective of the project was to provide an application that can predict the job placement status of a new applicant with an accuracy rate of at least 80%. The objective was met because an application interface was built that can take user input and predict whether the applicant is likely to be placed or not placed in a job. Upon evaluating the accuracy of the model, using an accuracy score metric, it is shown it was able to predict with at least an 80% accuracy rate.

Effective Visualization and Reporting

The visualizations help support the non-descriptive method because they show the contents of the dataset used for train and testing the machine learning model. It is important for end users to be able to visually see trends in the data before they enter in their own for an applicant. The histograms and scatterplot matrix can be quickly glanced at to see how many people are getting test scores within a specific percent range. If the user wants to dig deeper into the relationship of test scores to see which scores are the most important, they can analyze the scatterplot matrix because it shows a comparison of paired test scores for each test. If all the end user wants to see is a summary of the percentage of employable and non-employable individuals within the dataset, they can use the pie chart because it gives exactly those two pieces of information. If the user did not want to use the application to make the prediction, they would be able to reference the visualizations to determine a prediction on their own. They would just have to compare the scores of the applicant they are reviewing to the visualizations provided.

Accuracy Analysis

An accuracy score was used to evaluate the accuracy of the predictions, represented as a percentage, being made by the model. I was looking for an accuracy rating of 80% or better and the model can consistently perform at that expected rate. The rate could be better, but for the purposes of solving the problem I set out to solve, 80% is efficient enough.

For example, when a user enters in the test scores as 82 76 70 91 62, they will get a result of "Placed". Based on the model meeting the minimum requirement of 80% accuracy, we are confident that the result is at least 80% accurate.

Application Testing

The application was tested with multiple machine learning algorithms. I tried using linear regression, Gaussian Naive Bayes, and Support Vector Machines (SVM). Using the same input to train

each model, an accuracy metric was evaluated to see which model was consistently the most accurate when passing the test data through the model. Upon the evaluation of the best model, the other two models were discarded from the source code. Then, another test run was performed to make sure the model still performed as expected. I was looking for a prediction accuracy rating of 80% or better, so I also changed the test size that was used in the model. This allowed for me to see what test size would present more accurate predictions. Ultimately, I settled on 0.3 because it consistently met the 80% accuracy rate. This was done to ensure the most efficient model was being used for the application.

Application Files

All working files are contained in the GitHub repository located at <https://github.com/dreev65/WGU-Capstone>. The only file that needs to be accessed is the WGU Capstone.ipynb, however the csv file containing the dataset is present in the repository as well.

Order of files by importance:

- WGU Capstone.ipynb
- job_placement_data.csv ← file used in application
- unprocessed_job_placement_data.csv ← unprocessed dataset downloaded from Kaggle

All libraries used in the application are installed at the top of the application and imported directly under the installation code for each of those libraries.

Order of libraries by importance:


- The order of the libraries has no importance or impact on the application. If they are all installed and imported correctly, the application will run properly.
 - The included installations for libraries include the following:
 - Numpy
 - Pandas
 - Matplotlib
 - Sklearn

User Guide

Below are the steps required run the application. Nothing needs to be installed before execution. All libraries used will be installed upon executing the application.

To run the application:

1. Open web browser
2. In the search bar, copy and paste or type in the following URL:
 - a. Mybinder.org
3. Hit enter key on keyboard
4. In the “GitHub repository name or URL” text box, enter the following URL:
 - a. <https://github.com/dreev65/WGU-Capstone>
5. Click the “Launch” button

6. Wait for myBinder to direct you to the notebook launcher
7. Once on the notebook launcher, on the left side there will be two files listed in a file tree. Double click the “WGU Capstone.ipynb” file in that tree.
 - a. *Notebook should open to the right of the tree in the main window*
8. Next click the double play button in the top ribbon of the notebook window
 - a. Button looks like the following icon: 
9. Click the red “Restart” button
10. Application will start running (first thing to run are the library installs)

Entering input as a user:

1. Once the application has begun running, scroll down to the “User Interaction” section
 - a. Bottom of the application
2. Enter in any 5 test scores (scores are percentages) separated by a space
 - a. Example: 82 76 70 91 62
3. Hit the enter key on keyboard to continue application execution
4. Result will be shown below the input entry field
 - a. Above example should return “Placed”

Summation of Learning Experience

My previous course work readied me for this project because I was already familiar with the programming language I decided to use. Python was used in the Data Structures and Algorithms II – C950, which was beneficial to this project because I was able to understand and write the code a bit easier. The basic syntax was not something I had to relearn or learn for the first time. The knowledge obtained in Introduction to Artificial Intelligence - C951 was also helpful for this project. Particularly Task 3 was the most beneficial because provided an understanding of what some of the written documents should contain.

Even though the basic Python was recognizable and understandable, some of the libraries were libraries I had never seen before. Therefore, I had to do some research to determine what libraries would need to be used in the project. After that, I had to research and learn the syntax for those libraries in order to implement them with my project. I also had to research Jupyter Notebooks to learn how to use it for this project. Along with that, I had to research how to deploy the project for easy use, so I had to research different ways to access a notebook from a shared environment.

The technology industry is ever changing, so continually learning new things is an essential skill to have. Within the world of computers there are so many ways to do the same task and that became very clear with this project. I don’t see myself ever trying to stop learning new things because that is half the fun when working with computers. Some of the most enjoyable parts of the project were researching new and different technologies that could be used to implement the application. This project really drives a focus on going out there and using any resource possible to accomplish the goal due to the open-endedness of how the application can look and what the topic can be. It is easy to see that continue learning is beneficial in not only a computer science aspect, but in almost every aspect of life.

Resources

This section is intentionally left blank.