MINNESOTA STATE UNIVERSITY, MANKATO
DEPARTMENT OF COMPUTER INFORMATION SCIENCE
COMPUTER SCIENCE PROGRAM
HANDOVER DOCUMENT
THE BETTER BUSINESS BUREAU TEAM

Clients:

Ryan Sharp
BBB Twin Cities

Eli Johnson
IABBB

Rubens Pessanha
IABBB

THE BETTER BUSINESS BRIGADE

Justin Engels
Database Lead

Collin Dahlback
Testing Lead

Biruk Anley
Data Scraping Lead

Jackson Thoe
Algorithms Lead

Dylan Moore
Team Lead

FACULTY COACH

Mike Mansbach

Table Of Contents

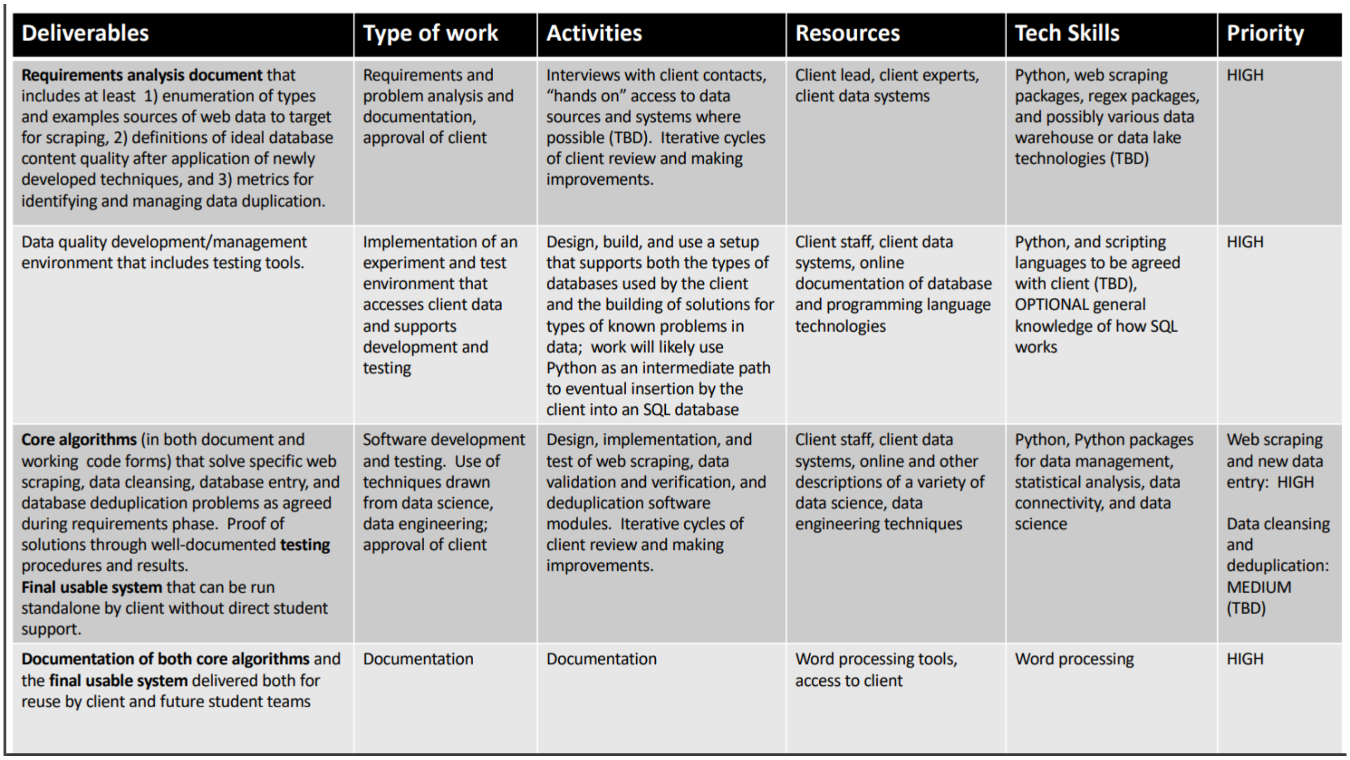
1. Introduction

The Better Business Bureau (BBB) is a highly respected non-profit organization that has been at the forefront of promoting ethical business practices and advancing marketplace trust for over a century. Its unwavering commitment to ensuring fair and honest dealings between businesses and consumers has earned it a reputation as the gold standard in business accreditation.

However, as businesses evolve and adapt to changing market conditions, it becomes increasingly important for the BBB to maintain accurate and up-to-date information in order to retain the trust of consumers and the business community. In response to this critical need, our project is to develop a sophisticated script that will gather new and reliable data each time it runs, while providing a confidence measure to ensure that the information collected meets the exacting standards of the BBB.

2. Getting Started

2.1 Deliverables



Deliverables	Type of work	Activities	Resources	Tech Skills	Priority
Requirements analysis document that includes at least 1) enumeration of types and examples sources of web data to target for scraping, 2) definitions of ideal database content quality after application of newly developed techniques, and 3) metrics for identifying and managing data duplication.	Requirements and problem analysis and documentation, approval of client	Interviews with client contacts, "hands on" access to data sources and systems where possible (TBD). Iterative cycles of client review and making improvements.	Client lead, client experts, client data systems	Python, web scraping packages, regex packages, and possibly various data warehouse or data lake technologies (TBD)	HIGH
Data quality development/management environment that includes testing tools.	Implementation of an experiment and test environment that accesses client data and supports development and testing	Design, build, and use a setup that supports both the types of databases used by the client and the building of solutions for types of known problems in data; work will likely use Python as an intermediate path to eventual insertion by the client into an SQL database	Client staff, client data systems, online documentation of database and programming language technologies	Python, and scripting languages to be agreed with client (TBD), OPTIONAL general knowledge of how SQL works	HIGH
Core algorithms (in both document and working code forms) that solve specific web scraping, data cleansing, database entry, and database deduplication problems as agreed during requirements phase. Proof of solutions through well-documented **testing** procedures and results. **Final usable system** that can be run standalone by client without direct student support.	Software development and testing. Use of techniques drawn from data science, data engineering; approval of client	Design, implementation, and test of web scraping, data validation and verification, and deduplication software modules. Iterative cycles of client review and making improvements.	Client staff, client data systems, online and other descriptions of a variety of data science, data engineering techniques	Python, Python packages for data management, statistical analysis, data connectivity, and data science	Web scraping and new data entry: HIGH Data cleansing and deduplication: MEDIUM (TBD)
Documentation of both core algorithms and the **final usable system** delivered both for reuse by client and future student teams	Documentation	Documentation	Word processing tools, access to client	Word processing	HIGH

2.2 Initial Information

At the outset of the project, we were provided with a sample database containing 160,000 rows, each representing a business, and 24 columns providing information about each business. However, not all businesses had complete data, with some lacking URL, email, phone number, or address information. We discovered that a significant amount of data was missing, with 88,043 URLs, 81,875 emails, 4,822 phone numbers, and 2,622 addresses absent from our dataset. Our initial objective was to prioritize the missing information and determine methods for retrieving it. We determined that our priority order would be URL, email, phone number, and address in descending order.

2.3. Skills Learned

Following discussions with our client, it was expressed that they preferred we utilize Python for the scripting portion of the project. Subsequently, we conducted a series of exercises to refresh our knowledge of Python and explored several Python packages that could assist us in analyzing and correcting data. During our data collection phase, we identified Beautiful Soup as an intuitive and efficient tool for parsing data, and thus determined it would be the most suitable option for our needs.

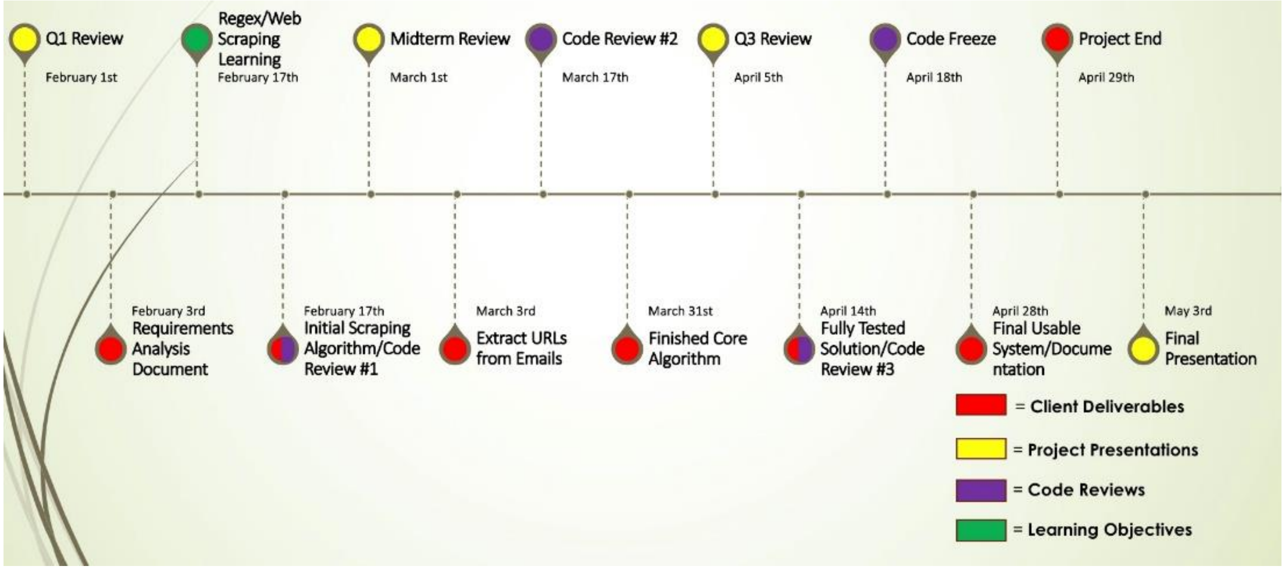
2.4 Documents created

There were a couple of documents we developed to get started with this project.

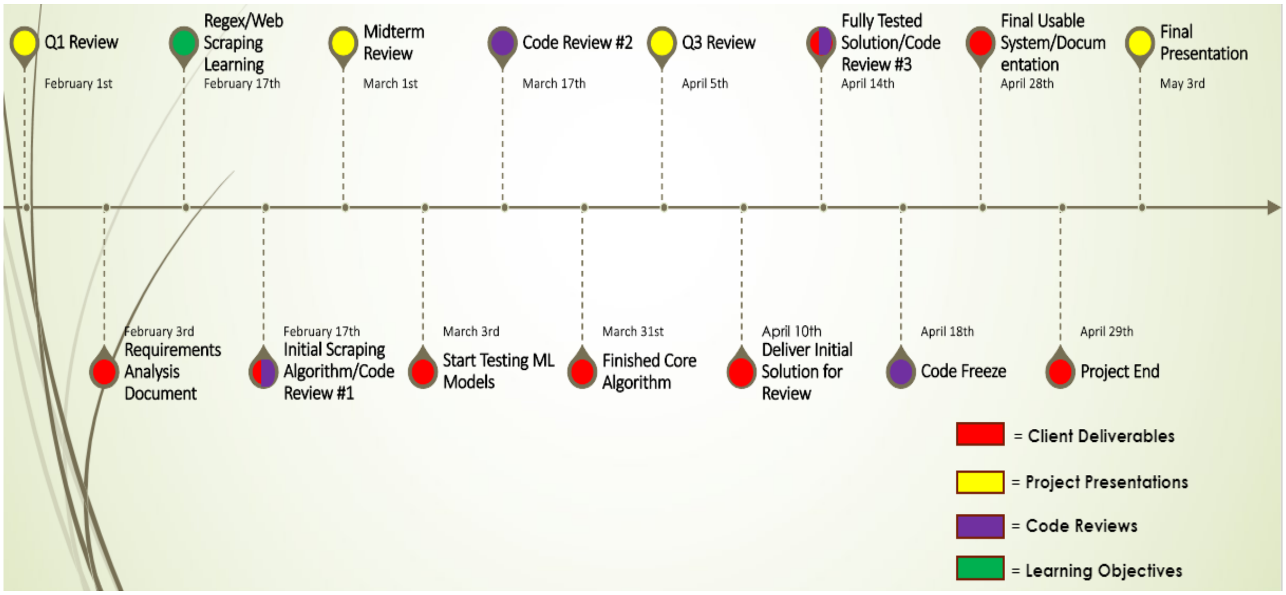
1. **Timeline and Milestones** – is a document that uses milestones to divide project work into phases for easy monitoring. This document includes a list of critical dates and actions included in the project.

Our timeline changed over the course of the project to reflect our current progress at any point in time. After the midway point in the project, we assessed our position and shifted deliverable dates within the timeline to show how we were ahead of schedule on certain aspects. We also added a section to show our expanding work on the machine learning models we were building at the time. These changes are shown below.

Old Timeline



New Timeline



2. **Requirements Analysis Document** – a document that includes a full description of all types of data and databases to be included in the project, together with an analysis of all (or most, as possible) of the known problems in the data, together with typical downstream consequences (business and technical) associated with each type of problem.

3. Documents and Code

This section will cover all folders within our final usable system, which in this case is our GitHub. It will go over what each folder contains with a summary of the general purpose of each folder.

3.1 Extract Data

The purpose of this folder is to contain all the scripts that do all the web scraping work. They search the web to create new URLs, extract data from a website, and run the independent variable functions on them. It also contains two csv files that store all that data and check which independent variables are the best for our machine learning model.

The documents within this folder:

README.md

1.) create_urls.py

2.) data_extraction.py

3.) fill_ind_var_columns.py

4.) filled_ind_vars.csv

5.) best_ind_vars.csv

3.2 Main Code Runners

The purpose of this folder is to contain main.py and all of the other main files. This is the primary folder where execution of our main script is run.

The documents within this folder:

README.md

1.) main.py

2.) main_url_scrape.py

3.) main_scrape_data.py

4.) main_machine_learning.py

3.3 Not Our Code

The purpose of this folder is to contain scripts that were not created by us that we have incorporated into our project.

The documents within this folder:

README.md

1.) elis_functions.py

2.) get_status_code.py

3.4 Data

The purpose of this folder is to act as the primary place of storage for most of the csv files we work with. It contains all the initial csv files we got from BBB's database, and some csv files that our scripts work with, including a ml_data subfolder which contains important data collected from our machine learning models.

For a full list of all data documents within the folder, see README.md

README.md

3.5 ML Models

This folder is a collection of files and scripts that pertain to the development, testing, and visualization of machine learning models. The primary purpose of this folder is to create and test machine learning models on various inputs, which can either be numerical or categorical. The folder contains different types of machine learning models, such as regression models and classification models. Additionally, the folder contains a visualization sub-folder that contains scripts for generating different types of visualizations such as bar charts, logistic regression plots, confusion matrices, and others. These visualizations can be useful for interpreting the results of the machine learning models and for communicating those results to others. Overall, the folder is a valuable resource for individuals and organizations that are interested in developing and testing machine learning models on different inputs.

The documents within this folder:

README.md

1.) decision_tree_classification.py

2.) dt_model.pkl

3.) logistic_regression_model.py

4.) lr_model.pkl

5.) Visualization sub-folder

I.) ClassificationVisualization.py

II.) RegressionVisualization.py

3.6 Testing

The purpose of this folder is to contain scripts that test several of our functions.

The document within this folder:

README.md

1.) test_create_urls.py

2.) test_fill_ind_var_columns.py

3.) test_get_html.py

4.) test_get_status_code.py

5.) test_scrapers.py

3.7 Tools for manual checking

The purpose of this folder is to contain scripts that help with manually checking websites for the machine learning models.

The documents within this folder:

README.md

1.) manual_url_checking_helper.py

2.) manually_check_scrapers.py

4. What We Explored

4.1 What Didn't Work

1. Linear Regression Model – We were initially going to use the linear regression model as our machine learning model, and while it can still be used it's the worst model by far.

2. Searching google – At one point we had one of our scripts mass searching google for data and we never found a good way to bypass the search limit. We tried proxies and multiple scripts to try to make the time spent reasonable, but we could never make it work.

3. Getting good independent variables – While we were able to come up with several independent variables, the process of finding them and making sure they're actually good and creating them and making them work properly was a long and painful process that produced mixed results and is still a huge area for improvement.

4.2 What Worked

1. Regex – We tried to use regex to scrape data off from websites, and as painful as regex is it ended up doing a good job and worked out in the end.

2. Decision tree – While the linear regression model didn't go very well, the decision tree ended up showing a lot more promising results and is the primary machine learning model we used.

3. Manually checking websites – We initially feared how long it would take to manually check 2000 websites, but thanks to selenium and a small script we made using it, it ended up being a very quick and easy process.

4. Incorporating the previous team's project – Considering how different this project was from the first project, we initially ignored it, but when we started looking into it, we found parts of it we could use, and we were able to easily incorporate their project into ours.

5. Areas for Project Expansion

5.1 Feature Improvement

The main area for improving this project would be investigating and creating new features (independent variables). Websites can be made thousands of ways and our scripts unfortunately won't give correct results for them all. A key expansion of this project would be finding additional ways to associate a website with a business. Linked here are the features we have explored during our time with this project. Please do add many more you might find insightful.

5.2 Scraping Improvements

While scraping the websites contained in BBB's database for new data, there exist lots of different types of websites (There is no one best practice for designing websites, this makes it hard to find patterns which store data). This makes it difficult to build scripts which work for each different type of website. We made sure our scripts covered the best practices for web design, but an improvement of this project is to discover outlier websites who may store their information in a different fashion and ensure the scripts cover those as well.

5.3 Verification and Purification of Data for Removal of Duplicates

While web scraping, we have collected a lot of data that we can't track if there is a duplicate or not. If we were to have more time to work on our data scraping project for the BBB, we would give priority to incorporating advanced computing components to ensure precise data deduplication and cleansing. This would entail utilizing potent algorithms and software tools tailored for detecting and eliminating duplicate data.

5.4. Check if a URL has been updated

Rather than running the script for all the businesses we can find ways to check if a business URL made an update. One way to do this is by using web scraping tools that can detect changes in the HTML code of a webpage. For instance, we could use Python libraries such as Beautiful Soup or Scrapy to compare the HTML code of a webpage at different times and identify any differences or changes. Alternatively, we could use online services that monitor and track changes to websites, such as Visual Ping or Distill Web Monitor. These tools can send you notifications when a webpage has been updated or changed, allowing you to stay up to date on any changes to the content of the webpage.