SP-2 DataMining & AI Software Final Report

4803 - 03

Fall Semester 2024

Professor Perry

2024

https://github.com/TerrorismAnalyticsBureau

https://terrorismanalyticsbureau.github.io

 Tanner Velzy Project Lead/Documentation	 Andujar Brutus Development

Name	Role	Cell Phone / Alt Email
Tanner Velzy	Team Leader/Documentation	404-405-3524 Tdv1201@gmail.com
Andujar Brutus	Developer	470-476-4473 andujarbrutus@gmail.com
Sharon Perry	Project Owner/Advisor	770-329-3895 Sperry46@kennesaw.edu

Lines of Code: 7,104

Number of Project Components/Tools: 19

Total Man Hours: 165

Contents

1.0 Overview

In this project, we are utilizing data mining and AI to analyze a dataset and display the data in a readable format as well as make a prediction based on the data. We will be using a CICD pipeline to streamline the process as well as handle version control. Azure will be our primary software from which we base our pipeline as well as do most of our work. Microsoft Fabric and PowerBI will be used in manipulating the data while Github Pages will host our frontend with lambda workstation bridging the frontend to the backend. All of this will allow us to display our findings from data mining and the AI's own predictions based on said data.

2.0 Requirements

2.1 Introduction

2.11 Overview

In this part of the report, we highlight the various requirements that our project, Datamining and AI, will need. It will go over the requirements in each assigned section which, by the end, should cover the entirety of the project requirements. The requirements should account for both the application and AI parts of our project. The requirements will not cover the specific costs, scheduling, and design as all of that is covered in other documents with the corresponding title.

2.12 Project Goals

- Develop a data mining application and data analysis AI that will both display collected data in a uniform fashion and find patterns within the collected data as it relates to crime of terrorism.
- Institute cloud software into the data mining application to make the application itself more efficient at collecting data from various news sources.
- Set up a CICD pipeline to make version control easier to manage between group members.

2.13 Definitions and Acronyms

CICD - Continuous Integration Continuous Deployment

AI - Artificial Intelligence

ML - Machine Learning

Scrum – a management framework that teams use to self-organize and work towards a common goal.

Yaml – readable data realization serialization language.

2.14 Assumptions

It is assumed that the user will have access to the internet and a non-mobile device such as a laptop or desktop personal computer. Users will have affiliation with a private organization that owns the application.

2.2 Design Constraints

2.21 Environment

Our data will be displayed on the Kennesaw website domain that is provided to each student. Azure will be our cloud software of choice that the data mining application will use. Our intended environment for our application and AI to work will be on a computer with an internet connection.

2.22 User Characteristics

Users that we predict will most commonly use this are those in law enforcement, government agencies, or private organizations. It can also encompass those who might be curious about crime correlations or statistics.

2.23 System

We are using the Python language to code our application and AI, therefore any machine learning libraries outside of Python's already existent libraries are out of our reach. We are also constrained to any of Azure's capabilities as free cloud software. There is also a limit we can reach with Azure as we are using a free account. The lambda workstation also holds a portion of our code due to its importance in our project.

2.3 Functional Requirements

Application and AI will retrieve and use the data to output into the website.

Dataset will be stored in cloud software, allowing it to be accessed from anywhere the application/AI runs.

Display Home Page of website.

The website itself will be in an easily readable format with an appealing look.

Short summary of what the user is expected to see.

Visualization of the data collected by the application

Will go over only what pieces of data we see as important enough to show.

AI's content is based on its findings for the data.

The AI will be a time-series analysis model or LSTM.

2.4 Non-Functional Requirements (use if applicable)

2.41 Security

The data set we decide to use will be anonymous and closed to the public as it will be synonymous with our application and AI.

2.42 Capacity

The website should boot up in a timely manner after the data has been processed through all parts of the application and AI. As we are using cloud software, the data will not have to be stored in the program itself and instead can be easily called on, which should make the program less storage intensive.

2.43 Usability

Our project should be usable for anyone using the internet as it will be all displayed on a website. The website will also be user-friendly, allowing a user to easily view the data and see what conclusions the AI has made. The user will also have to have permission to view the data within an organization.

2.44 Other

The website will be available to anyone and anywhere with a working computer whether that be a laptop or personal computer. With our cloud software, the website should be able to access the dataset it needs with ease, allowing for shortened loading times no matter where the user accesses the website.

2.5 External Interface Requirements

2.51 User Interface Requirements

We will be utilizing Github Pages to host our website which will display the mined data and AI findings. The website will be easily readable and only accessible by those who have permission.

3.0 Analysis

3.1 Terrorism Dataset

The dataset we have decided to use is the Global Terrorism Database from the website Kaggle that shows attacks from 1970 to 2017. The dataset has information on more than 180,000 terrorist attack within a csv file. The file comes with 135 columns of different pieces of information from the date the attack was committed, the kind of attack it was, and even the victims and perpetrators of the attack.

From the dataset, we will extract certain columns of information that will give us the best profile of each crime. These columns are the year, month, day, country, region/city, attack type, target type, target, weapon type, terrorist or terrorist group, and how many casualties as a result. These pieces of data will provide the best profile for the terrorist attack and what would be best used by the AI.

3.2 Azure

Azure is a cloud platform that provides cloud storage, DevOps, Machine Learning Ops, and other resources that an organization can utilize. It can fill the role of our data warehouse and hold the dataset so that other parts of the pipeline can access it and retrieve the data they require for their task. Provides virtual machines and other computing resources for computations. Used Azure's repos to facilitate a CICD pipeline which can be connected to other software.

3.3 CICD Pipeline

CICD, or Continuous Integration Continuous Deployment, is a software development process that allows for constant building, testing, and deployment. In a pipeline, it automates this entire process throughout our project, ensuring that we have less work regarding building, testing, and deploying the code ourselves. This increases our workflow significantly and allows us to push out updates for our project much faster and assist with version control as it will automatically update during a time we set.

4.0 Design

4.1 Introduction and Overview

Here, we will describe the design of our project's systems. This section will provide a high-level overview of the design and architecture of our system as well as our approach to this project. This will also explain why we plan to use these designs and approaches.

4.2 Design Considerations

4.3 Assumptions and Dependencies

- Client requires a working PC with an internet connection.
- Proper authorization for Azure and GitHub
- Application and AI require an informative dataset.
- Depends on Azure functioning correctly.
- Cloud software must be online, and the user must be connected to internet to allow the program to access the dataset.
- The Lambda server keeps up with requests and data.
- The CICD Pipeline functions correctly and without issue

4.4 General Constraints

- Dataset accuracy
    - This entire project depends on the dataset being accurate, which we will do our best to circumvent by using a trusted source.
- Cloud software uptime and downtime
    - As it is an online service, our storage is based on the uptime and downtime of our provider.
- Python library limitations
    - The limitations are based on the version of the libraries themselves depending on the version of Python we use.
- Azure free cloud software storage provided.
    - Due to being a free version of the cloud software, the storage might be limited. has only a limited number of credit hours before asking you to begin paying for their service.
- Effectiveness of the automated testing
- Budget
    - We will be using all free options with any software we decide to use, which might limit the capabilities of what we might be able to do with paid subscriptions.
- Lambda Server limits
    - Has a limited amount of processing power compared to cloud software.
- CICD pipeline constraints
    - As it is running constantly, resource management will be a big part. This would mean either watching over the pipeline or automating it completely.

4.5 Development Methods

We will be using Scrum, of which we will be using Continuous Integration & Continuous Delivery (CICD). We are using Scrum because it is a framework for managing work that will allow us to work together efficiently and focus on a common goal. We decided to use CICD because it will allow us to keep up with the version control of the project as we develop it. This will help automate testing, training and publishing while ensuring the group always has the same version.

4.6 Architectural Strategies

- Automated Deployment
    - We will be designing the architecture API using an IDE known as PyCharm. We will then use the IDE/Terminal to connect to a version control repository that is connected to GitHub to allow us to use CICD. Using GitHub actions, we will create a CICD pipeline to use with Azure. This pipeline will be responsible for testing and training of the application and AI.
- Azure
    - A cloud software with a free version that we will be using to act as a data warehouse for the dataset we decide to use for our application and AI. Our second

choice was Google Cloud because it is convenient, but Azure was more lucrative for job opportunities as more companies use it compared to Google Cloud.

- GitHub
  - It is a developer platform that will allow us to store and share code which we will be using for version control and to store the code we have made in a way that is easily accessible by the group. GitHub would also allow us to make the most of CICD due to its usability with group projects and documentation, however, due to budget constraints, we will not be able to utilize CICD and instead must base it on Azure instead.
  - Github Pages will also be used to host our frontend. It is a free static website hosting service created from Github repositories. With this, we can easily and for free host our website. This will allow for us to integrate our frontend with our pipeline and allow for the pipeline to automatically update the frontend along with the rest of the project.
- PyTorch
  - A machine learning library that will help train the AI and create the architecture for the AI. We were also considering using TensorFlow as it is used for big projects in companies, but PyTorch was more suited for smaller projects like ours.
- Python
  - Programming language that is better suited for our machine learning objectives.
- Terrorism Dataset
  - This dataset is taken from the Kaggle website which will detail terrorism-related crimes that have happened between 1970 to 2017. It has 135 columns of data; however, we will be using only columns of data that are the most important to display and for our AI to digest.
- PowerBI
  - An interactive data visualization software product that allows us to more easily display necessary information from the data mining side of our project.
- OneLake
  - A data lake that will act as our data repository where we retrieve data for Microsoft Fabric to use.
- Microsoft Fabric
  - An end-to-end analytics and data platform designed for enterprises that require a unified solution. We will be utilizing this for the data lakehouse and OneLake.
- IntelliJ
  - By JetBrains, this functions as our IDE where we do most of the coding before implementing into the pipeline.
- Google Cloud
  - A free cloud platform that provides various cloud computing services. For our project, we utilized to design the AI using their resources.
- Microsoft Sharepoint + OneDrive
  - A cloud-based platform that allows a user to store, organize, and share files, information, and applications. For our project, we used its drive function to store our dataset for the AI to use, saving us money.

4.7 System Architecture

There are five subsystems in our system architecture: the Cloud, the Application, the AI, the Website, and the Bridge. The Cloud will store the dataset we have given it so it may be accessed more easily by the Application. The Application will download the dataset from the cloud and process it into various categories and patterns. The AI will then take this data that the Application has processed and draw conclusions from the data. Both the Application and AI will submit their findings to the Website where it will all be displayed in an easily readable format for the average viewer. And finally, the Bridge connects the backend of the systems to the frontend as the frontend will request data using the Bridge.

4.8 Detailed System Design

4.81 Classification

- The Cloud - Subsystem
- The Application - Backend
- The AI - Module
- The Website – Frontend
- The Bridge – Server

4.82 Definition

- The Cloud – The main goal of the Cloud is to store our dataset that the application and AI will use.
- The Application – Will communicate between the cloud and backend as well as gather and evaluate the data for use by both the AI and Website.
- The AI – The AI will process the dataset provided by the Application and correlate any patterns from the data before outputting them into the Website.
- The Website – The dedicated frontend that will display all the outputs from the Application and AI.
- The Bridge – This acts as the bridge between the frontend and the backend, or everything and the Website.

4.83 Constraints

- The Cloud – A potential constraint for the Cloud is the fact we are using a free license that could limit the processing power available to us.
- The Application – Potentially, there could be a compatibility between the Application and the Cloud that could prevent us from even using the dataset.
- The AI – Depends on the Application for its data so it will not even get the dataset if the Application doesn’t either. There is also the possibility of the data being wrong which is more important for the AI to have correct data as it is making assumptions and correlations on that data.
- The Website – Depending on how the data from the Application and AI is given, the Website might have issues formatting the data in an easily readable format.

- The Bridge – Constrained by the load the server can take and how many calls it is able to process.

4.84 Resources

- The Cloud – Has usable storage of 50 MB which should store the dataset we provide it.
- The Application – Utilizes the Cloud's data warehouse to retrieve the dataset that it will use. The Application will also be utilizing Python's libraries.
- The Bridge – Utilizes a server with the capability of transferring requests.

4.85 Interface/Exports

- The Cloud – Provides a data warehouse and technical dashboard for our machine learning needs. Azure services provide various machine learning tools and can be classified as a subsystem to our project.
- The Application – Provides communication between the frontend and backend while also processing the dataset it is given for display and use. This will use a module API that will communicate between our subsystems.
- The AI – Uses created model and dataset to provide time series analysis and anomaly detection using Azure Studio. This will provide a collection of files that will be exported to the Website for viewing.
- The Website – Displays the information created by the Application and AI for the clients to read in an easily accessible and readable format. This is a function that will allow us to display all the information about the dataset.
- The Bridge – Our server, which is on the lambda workstation as an API module, will take various GET calls from the frontend which it will process through the modules before giving that GET call to the backend at which point the frontend will receive the data it requires.

4.9 Design Glossary

CICD – Continuous Integration & Continuous Delivery; an automated process used by software development teams to streamline development, testing, and delivery of products.

Scrum – A collaboration framework used in software development and other industries that helps group's structure and manage their work.

Cloud Software – Software that is based in the cloud which in turn operates over the internet rather than a defined physical space, allowing people to store and access data from anywhere with an internet connection.

IDE – Integrated Development Environment; a software that provides development tools and a GUI for programmers.

Data Lakehouse – creates a single platform by combining key benefits of data lakes and data warehouses.

Data Lakes – Large repositories of raw data in its original form.

Data Warehouses – organized sets of structured data.

Lambda Workstation – serverless computing service.

5.0 Development

5.1 Development Set Up

Starting from the beginning, the first step was to find a good dataset which in our case was the Kaggle Global Terrorism database. The more information for our project the better and this dataset provides it. The next step is to create an Azure Cloud account. This is where we will be holding the dataset as well as where most of our work will be done. In this case, we used our school account for this. After an account for Azure is made, the next step is to set up Azure DevOps from which the CICD pipeline will begin. After that, Azure is linked to Github which will automatically back up Azure DevOps to Github.

Once that is done, we set up our Machine Learning Ops repository in preparation for creating our AI once we get to it. Then we set up Microsoft Fabric and within it, set up PowerBI on the backend as well as OneLake. PowerBI will be used on the backend and frontend while OneLake will act as our data lakehouse where the data can be more easily manipulated. The next step is then to set up Lambda Workstation as it will bridge between our backend and frontend. We also created API modules in the Lambda Workstation to make this process easier. Finally, our set ends with creating the Frontend using Github Pages and Jekyll.

5.2 CICD Pipeline

As was discussed before, the CICD pipeline is how we managed both version control and run the entire project. It is made to be based in Azure DevOps where it branches out to Github and other parts of the pipeline from the backend to the frontend. The pipeline allows us to update the projects automatically every Friday ensuring we never fall behind on versions and allowing us to focus on coding. The pipeline connects all parts of the project, so it is safe to say that it is one of the most important parts of the project. The pipeline begins in Azure where the DevOps and it functions as the Continuous Deployment part of CICD.

Continuous Integration involves building, testing, and merging. In our project, Azure provided this service and updated the version every Friday. It would back up the new version to Github repository before pushing it throughout the rest of the pipeline, which is where the Continuous Delivery part of the pipeline commenced. This would update the rest of the pipeline and ensure every part of it was the same version as Azure's.

5.3 Azure

Azure acts as our main repository, pipeline foundation, and IDE when required and not using IntelliJ. Everything regarding the project can be done here for both data mining and AI. Azure also provides Microsoft Fabric which provides PowerBI and Microsoft Fabric, the former being our frontend and the latter being our data lakehouse where we can manipulate the data. Azure also functions as our cloud software, allowing us to store data for use anywhere.

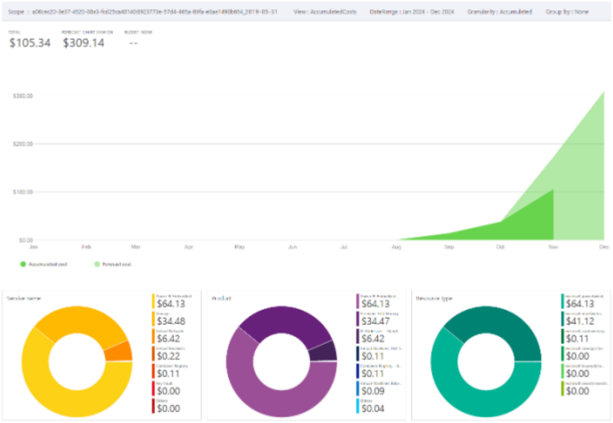


Figure 1.

This is also where we build the AI, however we ran into the problem of our Azure free trial of credit hours running out which is shown above in Figure 1. This issue was caused when trying to change the source of the dataset to Azure Storage which caused PowerBI to cost credit when extracting data from the dataset. While this is what is supposed to happen, we couldn't pay for the upkeep which caused us to max out our credit hours. While we were able to finish most of it, we were limited by our free account on Azure.

However, we found a way to get past this limited credit issue and are still able to use Azure without having to transfer everything to another platform which would take time. As each school account automatically gets 100 credits upon signing up, we signed up with another school account as we had another person in our group that could use their school account. While we had to transfer everything over and set up a few things which took only a little time, it worked flawlessly and allowed us to continue working on the project using Azure without using up the credit the way we did with the previous account. To avoid the issues that caused us to max out our credit on the last account, we stored the dataset in Microsoft Sharepoint which wasn't our ideal solution for an enterprise problem, however for the sake of our project it allowed us to store the dataset without costing us our credits on Azure.

5.4 Lambda Workstation

The lambda workstation is what bridges our cloud to the frontend and makes this whole project work. We did this through a variety of GET calls that ferry communications between the two sides. API modules were created to make this process much easier. The workstation itself acts as a server that will request data from the cloud to use in the frontend. It will specifically request from the PowerBI backend which itself will have received data from the OneLake data lakehouse.

5.5 Microsoft Fabric

Microsoft Fabric functioned as our data lakehouse, taking data and allowing us to manipulate it. It pulled from OneLake, a data lake, where we put the data from our dataset. OneLake will hold the data while Microsoft Fabric takes what it requires as set by us. From there, it will give the data to PowerBI backend which will sort the data as it awaits a call from the lambda workstation.

5.6 PowerBI

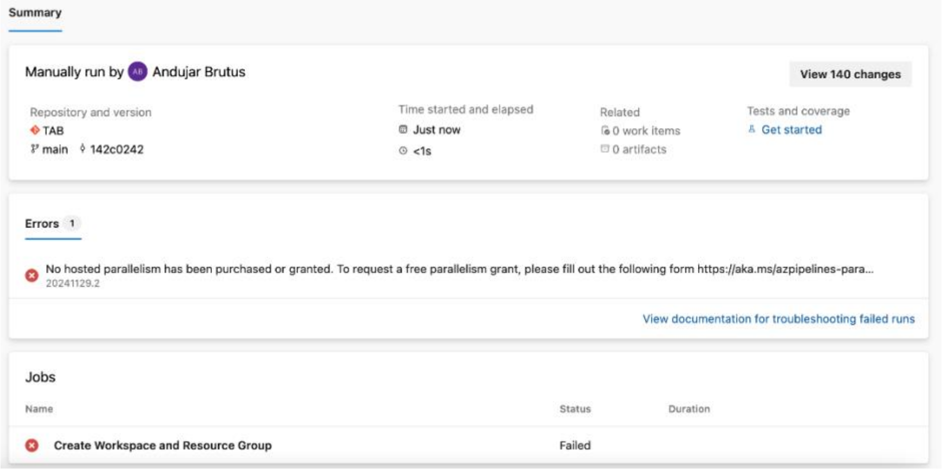
PowerBI played two parts in our development, backend and frontend. On the backend side, PowerBI would sort the data given to it at which point it will give the requested data to the workstation. Once at the frontend, PowerBI Embed will display the data in an interactive and easily readable format for the user, allowing them to manipulate certain parameters based on the data we have mined.

5.7 Google Cloud and Microsoft Sharepoint

As mentioned in 5.3, we got around the AI issue by both using another Azure account and using Microsoft Sharepoint instead of Azure to store data for our AI and PowerBI, saving money and not using our Azure free credit. In tandem with this, we used Google Cloud to design our AI. To do this, we used a service of Google Cloud called Google Colab. This allowed us to use Google's cloud resources to design the AI which both allowed us to not be forced to use our slower hardware and it is free to use. Learning from our mistake, we eliminated the need to use our Azure free credit while still getting the output we desired.

5.8 Issues and Problems/Abandoned Parts

As mentioned in 15.3, we had an issue regarding Azure's credit given to the account we were using, but we were able to easily fix this using another school account. If this didn't work, we would have had to port everything to Google Cloud which would have taken time and a lot of effort as we would have to learn how Google Cloud works and how to reintegrate the pipeline.



Manually run by AB Andujar Brutus

Repository and version
TAB
main 142c0242

Time started and elapsed
Just now
<1s

Related
0 work items
0 artifacts

Tests and coverage
Get started

No hosted parallelism has been purchased or granted. To request a free parallelism grant, please fill out the following form https://aka.ms/azpipelines-para...
20241129.2

Jobs

Name	Status	Duration
Create Workspace and Resource Group	Failed	

Figure 2.

A smaller issue we ran into when doing the AI was the ML Ops on Azure. To use it, we had to get permission from the Azure company to use parallelism to utilize ML Ops. However, we did not get permission in time as seen in figure 2, so we instead designed the AI using Google Cloud which allowed us to test it on Google Cloud using their resources as mentioned in section 5.7. Once ready, we transferred the code to a PI file that would then be stored on lambda workstation. We also provided the algorithm with its own dataset and connected the AI model to the endpoint that displays the output. While less convenient and requiring a few more steps, it serves the same purpose and provides the wanted outputs.

6.0 Testing

We tested 4 major parts of our project: the CICD Pipeline executing successfully, the Lambda Workstation/API working correctly and executing calls, being able to access the website, the data brought over and displayed successfully, and finally the AI giving a prediction based off the data given.

The reason each of these are considered major is primarily to do with how important they are to the functionality or idea of our project. The CICD pipeline is what allows all of this to work in the first place so without it, the project would not even begin to run. The lambda workstation bridges the frontend to the backend so without working, we would not be able to display our data. Getting onto the website is important as that is the main point of us showcasing our data. Ensuring the data is properly brought over is important as well since we want to stay as accurate as possible regarding the dataset we are using. Finally, having the AI use this data to predict potentially future terrorist acts will be our final major test relating to our project.

Below is a table showing our major and minor tests.

Requirement	Pass	Fail	Severity
CICD Pipeline executes successfully	Yes		Major
Lambda workstation/API works correctly, executing calls	Yes		Major
Get onto website	Yes		Major
Website format is correct	Yes		Minor
Data brought over successfully and displayed	Yes		Major
Look through the data with different parameters successfully	Yes		Minor
AI gives a prediction based on the data	Yes		Major

The links for the group members and Github works	Yes		Minor

Table 1.

In table 1, the AI part was a work in progress because, as mentioned in section 15.2, we ran out of free credits for our Azure subscription. However, as mentioned in 15.2, we were able to find a work around for both Azure and the AI with Microsoft Sharepoint and Google Cloud, allowing us to successfully test the AI, successfully completing all our testing.

7.0 Version Control

7.1 CICD

To best keep up with the versions and changes, we decided to use CICD or Continuous Integration Continuous Deployment. We will explain each part starting with Continuous Integration or CI. CI periodically integrates all the changes we made to the code to the main source code. This will also automatically test the code changes when the integration happens and therefore create a new build of the code. This allows us to keep up with new versions as well as allowing testing to occur automatically when the changes are integrated with the source code.

Continuous Deployment or CD works alongside CI to automate the release process of the application. After CI has integrated the code changes successfully, CD automatically deploys the application. This eliminates the need for one of us to go in and manually deploy our new application. This allows us to stay on schedule for any new builds as we can focus on getting the new build finished and allowing CD to do the final stages.

8.0 Conclusion

For Data Mining and AI, we learned about CICD pipeline as well as data mining itself. Each part of the pipeline is important from the DevOps all the way to the backend. We also came to learn about the fact of knowing your limits, especially when we reached the AI part of our project where we ran out of the free credits Azure provides. Fortunately, we were able to get around this using various platforms and a new Azure account which allowed us to fully deliver on our project. From this, we learned to look for different avenues of solving issues as some solutions might be easier than we think.

In conclusion, we learned that there are multiple avenues of using datamining and AI. From using a pipeline to automate the process to using many different resources and platforms to facilitate datamining and AI, this all allowed us to mine data, display pertinent information, and finally use AI to give its own prediction based on the data.

9.0 Appendix

9.1 Project Plan

Project Overview / Abstract (Research)

Our main objective with Data Mining and AI will be to find patterns in criminals using a combination of data and AI. To do this, we will be using cloud software in tandem with our datamining application to extract specific data from a dataset. The dataset we will be using is based on global Terrorism 1970 to 2017 with specific columns defining aspects of the crime The specific data, such as cities, careers, wealth, will then be displayed for reference before being inputted into our AI that will then use the data to find some pattern or correlation between terrorists. The AI, using the data it analyzed, will enhance crime statistics using data analytics and machine learning.

Project website

https://terrorismanalyticsbureau.github.io/

Deliverables - Specific to Your Project

1. A report of our data-based findings that our application extracts.
2. A report of our AI's findings in terms of correlations it analyzes.
3. SDD of both the application and AI.

Milestone Events

#1 SRS/SDD - By September 1

#2 Prototype - By October 8

#3 Final Report Package - By December 2

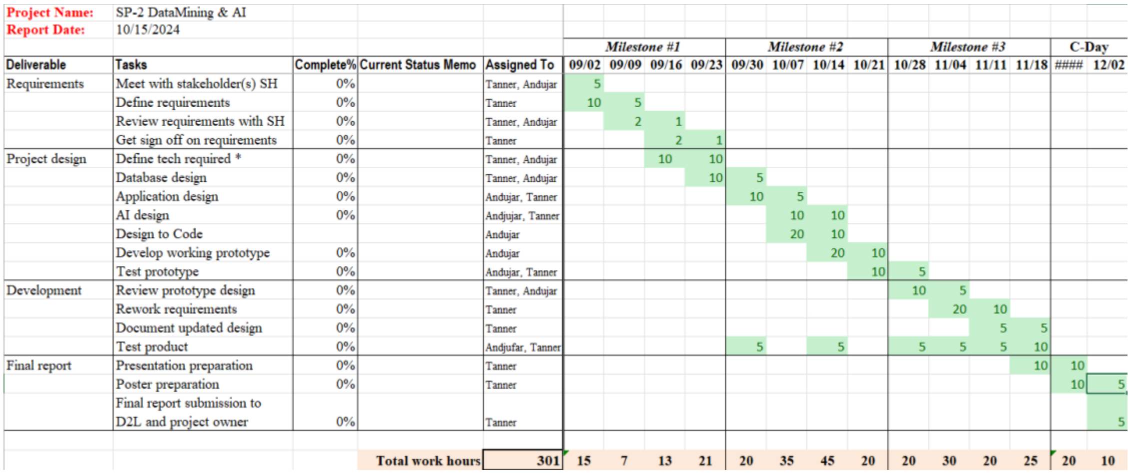
Meeting Schedule Date/Time

Biweekly every Thursday at 2:45 PM on Microsoft Teams.

Collaboration and Communication Plan

We will be meeting in Discord Monday and/or Fridays depending on what work needs to be done or we need to clarify some things.

Project Schedule and Task Planning



Project Name:	SP-2 DataMining & AI																	
Report Date:	10/15/2024																	
					Milestone #1				Milestone #2				Milestone #3				C-Day	
Deliverable	Tasks	Complete%	Current Status Memo	Assigned To	09/02	09/09	09/16	09/23	09/30	10/07	10/14	10/21	10/28	11/04	11/11	11/18	####	12/02
Requirements	Meet with stakeholder(s) SH	0%		Tanner, Andujar	5													
	Define requirements	0%		Tanner	10	5												
	Review requirements with SH	0%		Tanner, Andujar		2	1											
	Get sign off on requirements	0%		Tanner			2	1										
Project design	Define tech required *	0%		Tanner, Andujar			10	10										
	Database design	0%		Tanner, Andujar				10	5									
	Application design	0%		Andujar, Tanner					10	5								
	AI design	0%		Andjujar, Tanner						10	10							
	Design to Code			Andujar						20	10							
	Develop working prototype	0%		Andujar							20	10						
	Test prototype	0%		Andujar, Tanner								10	5					
Development	Review prototype design	0%		Tanner, Andujar									10	5				
	Rework requirements	0%		Tanner										20	10			
	Document updated design	0%		Tanner											5	5		
	Test product	0%		Andjufar, Tanner					5		5		5	5	5	10		
Final report	Presentation preparation	0%		Tanner												10	10	
	Poster preparation	0%		Tanner													10	5
	Final report submission to D2L and project owner	0%		Tanner														5
			Total work hours	301	15	7	13	21	20	35	45	20	20	30	20	25	20	10

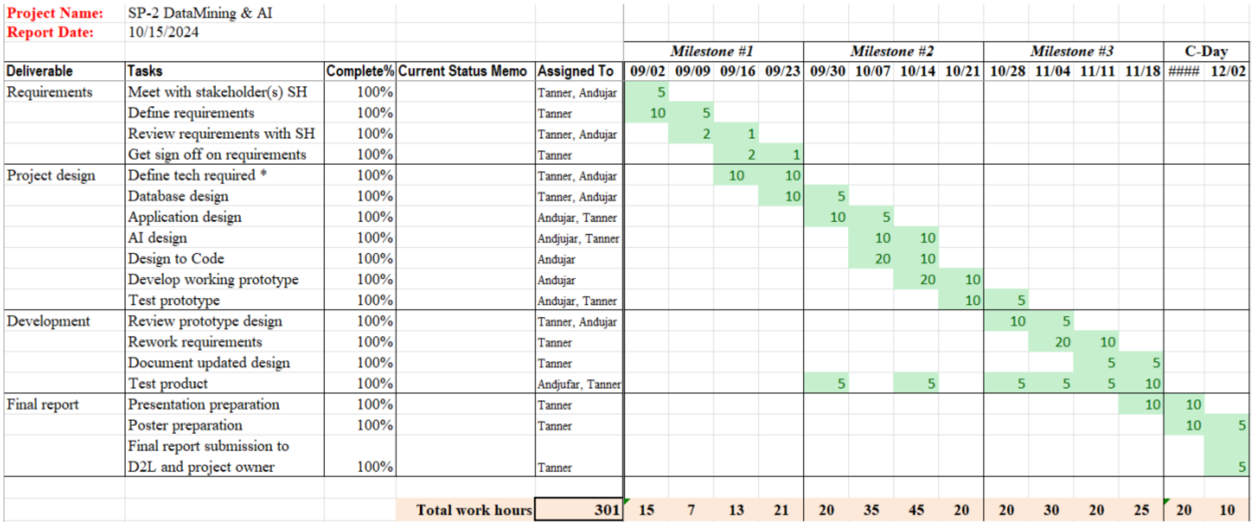
Other Plans – Like Risk Assessment (if applicable)

Along with the application and AI, we will be looking into data visualization to better showcase the data we have extrapolated. If we have time, we will look to compare different cloud software to see which one gives us better results.

Version Control Plan

We will set up a CICD pipeline, pushing every new change into our group GitHub so that we will stay on top of any updates we make and not accidentally use an old version.

9.2 Gnatt Time Chart



Project Name:	SP-2 DataMining & AI																	
Report Date:	10/15/2024																	
					Milestone #1				Milestone #2				Milestone #3				C-Day	
Deliverable	Tasks	Complete%	Current Status Memo	Assigned To	09/02	09/09	09/16	09/23	09/30	10/07	10/14	10/21	10/28	11/04	11/11	11/18	####	12/02
Requirements	Meet with stakeholder(s) SH	100%		Tanner, Andujar	5													
	Define requirements	100%		Tanner	10	5												
	Review requirements with SH	100%		Tanner, Andujar		2	1											
	Get sign off on requirements	100%		Tanner			2	1										
Project design	Define tech required *	100%		Tanner, Andujar			10	10										
	Database design	100%		Tanner, Andujar				10	5									
	Application design	100%		Andujar, Tanner					10	5								
	AI design	100%		Andjujar, Tanner						10	10							
	Design to Code	100%		Andujar						20	10							
	Develop working prototype	100%		Andujar							20	10						
	Test prototype	100%		Andujar, Tanner								10	5					
Development	Review prototype design	100%		Tanner, Andujar									10	5				
	Rework requirements	100%		Tanner										20	10			
	Document updated design	100%		Tanner											5	5		
	Test product	100%		Andjufar, Tanner					5		5		5	5	5	10		
Final report	Presentation preparation	100%		Tanner												10	10	
	Poster preparation	100%		Tanner													10	5
	Final report submission to D2L and project owner	100%		Tanner														5
			Total work hours	301	15	7	13	21	20	35	45	20	20	30	20	25	20	10