Oregon State University
College of Engineering

CS Capstone Technology Review

Fall Term
November 21, 2017

Code3 Visionary

Prepared for

Levrum Data Technologies

Carl Niedner _____ _____
Signature *Date*

Prepared by

Group 3

Kien Tran _____ _____
Signature *Date*

Abstract

This document will be outlining the pieces of technology the team is considering to be used in the project. The three main pieces that we will be discussing are different machine learning algorithms, types of databases, and types of application programming interfaces (API). The document will summarize the main reason for using these pieces and discuss the technology that is currently available for the team to use.

CONTENTS

1 Introduction

Code3 Visionary is a application that utilizes artificial intelligence and machine learning algorithms to predict future emergency incidents. The end product will be a web base application that would allow users to input a location value to receive future projections of the surrounding area based on past data. Our Capstone team is responsible for the development of Code3 Visionary from beginning of the application up to the end user interface. Although my partner and I will be discussing different technology components that we will be using for this project, each piece of the overall application will be done by both team members. My role is to understand the three different components that will be used in this project. These include machine learning algorithms, databases, as well as an application programming interface (API). Each piece of technology plays a vital role in the development of the application and decisions in the different options must be taken into consideration to optimize development.

2 Machine Learning Algorithms

2.1 Overview

Machine learning is the ability for a computer to learn using what it has been program to know. Broadly speaking, there are 3 types of ways a computer can accomplish this. The first is known as supervised learning where computers are fed information by users, in order to generate a an outcome. Using this method, the accuracy of learning can be recorded and shape to the desire of the developers. The second method is known as unsupervised learning. Here we do not have a known outcome for the machine to predict but rather seeing what the computer will predict from what is is already given. This method is great to understand clustering and seeing general trends. Finally, the last method is known as reinforcement learning. This method is training the computer to make specific decisions and allowing the computer to run by trial and error examples continuously in a controlled environment until it is capable of using it's past experiences to accurately make future predictions **?**.

2.2 Criteria

For our project, we will be working with the first method where we try to train a computer to make a predictive outcome and measure its accuracy. Following this method, we can tweak our training to see where we can improve upon to achieve higher accuracy for our predictions. The key idea idea for us is to enumerate the possible platforms we have available and evaluate their performance to achieve best results. The platforms that we decided to focus on are Apache Spark/MLlib, Google TensorFlow, and Azure. For each platform we want to see how well they can incorporate with our current libraries such as R and Python. We want to know if there are any upfront costs (both direct and indirect) that will need to be considered. We are interested to see if the options are open source and if there are any APIs available. By being open source we can gauge its level of custom development as well as the amount of support the option has with the community. The API support is another metric we will use to gauge ease of use.

2.3 Potential Choices

2.3.1 Apache Spark

Apache Spark/MLlib showcases many defining features that helps it standout in our assessment. From its documentation, it seems to be usable in a variety of languages including Java, Scala, Python, and R. Being open source they provide the software for free and have a community that can assist if we were to run into any issues. They also provide

a variety of machine learning algorithms including classification, decision trees, and clustering that we may find useful in our project **?**.

Some drawbacks of using Apache Spark is that there is no support for real-time processing, latency, and high memory usage. Since data is divided into batches of pre-defined interval, each batch of data are processes using operations like map, reduce and join. As a result these operation will return in batches hence why Spark does not support real-time processing. Its latency is also a contributing factor and is known to be an issue in the community. Finally, its usage of "In-memory" can be consider a bottleneck in the case of processing big data. Spark requires lots of RAM to run which is why it can be consider expensive **?**.

Pros: Free, Open-Source, Easily integrated with our desire languages, variety of machine learning algorithms

Cons: Not in real time, Expensive to use due to high memory usage by the program, High latency

2.3.2 Google TensorFlow

Google TensorFlow is open source as well as being develop by the tech giant Google give sit an edge in terms of documentation and wide adoption. Being open source their software is free to use and because of their community and backing, their is a strong indication that the software will constantly be updated. Its architecture allows us to deploy the computation not on just the CPU but on the GPU of our computers/server if we ever need to. They also offer installation on a variety of platforms including Ubuntu, macOS, Windows and other sources. They have API documentation that is compatible with Python which will make it easy for us to use **?**.

That being said, it does have some issues. Being built by a tech giant like Google, TensorFlow is very dense in terms of trying to find the correct information/algorithms we may need. It will take time to research and learn about TensorFlow before actually being able to dive in and use its libraries to use. Its libraries are written for Java, C, and Go which means we would need to be using API calls in order to access the functions we will need.

Pros: Free, Open-source, Computation can be done on variety of processor, Has API integration with Python

Cons: Dense (in terms of information), Libraries are in a different language (not one of our preferred ones), Research intensive.

2.3.3 Azure

Azure developed by Microsoft is unique in its approach to machine learning. It is browser-based utilizing a drag-and-drop environment to develop (no coding necessary). All development would be cloud base which means easier accessibility and development as a team. Their documentation and examples are extensive allowing for easier training and adoption of the program. It has a large number of machine learning algorithms that allows for training models in order to create predictive output which is what we hope to do. Their environment also supports current R and Python users using built-in packages that supports custom codes **?**.

The drawbacks seems to be mainly coming from its implementation. Since it is cloud based, a negative can be drawn about needing access to the internet in order to develop. It also has two version, a free and paid. The free version has a performance hindrance so purchasing of a license is required if more processing power is needed. The cost of the paid version is a per monthly/hourly charge based the number of seats needed and the number experimentation hours that will be needed. As of this report, Microsoft charges $9.99 per seat and $1 per studio experimental hours. Being a drag-and-drop model, it will also need special/extra code incorporated in order to catch cases where the model could not.

Pros: Cloud base means better collaboration, drag-and-drop makes for easier to program, strong documentation, support for R and Python.

Cons: Upfront cost, Cloud base means constant access to internet is require, Drag and drop will not have all the answers.

2.4 Discussion/Comparison

When comparing the Apache Spark and TensorFlow, they both have very similar aspect such as being open source, free, and containing an abundance of machine learning algorithms. That being said, they do differ in the language that was used to develop the backend which can resort in difference in speed and compatibility. Azure, being a drag-and-drop as well as cloud base, provides a different platform to develop application which may or may not be hinder it. Shown, is the table contain all three options and the criteria they satisfy in our check.

Factors	Apache Spark	Google TensorFlow	Azure
Price	X	X	✓
Language Compatibility	✓	X	✓
API Intergation	✓	✓	✓
Open Source	✓	✓	X

(Table 1): Showcases the comparison between the 3 different Machine Learning Algorithm software

2.5 Conclusion

Looking at the list we may see that Apache Spark fills out our criteria very well but it does have its drawbacks. Being developed from an interpreted language means that Apache Spark is slow and has a lot of overhead that could lead to expensive memory usage when preforming complex functions. Azure does provide a unique framework for the criteria we ask for, however, due to the upfront charges we may want to stay clear if there is a free software that can do what Azure can already. For this reason the Google Tensorflow would be the appropriate software the team should use. Even though its back end language is primarily done in C, this provides a faster user experience in terms of computational speed. The software is dense, so there will a lot of functions available for us to implement. Finally its API integration and being open source will allow us to find more ways to incorporate the software into our developing environment.

3 DATABASE

3.1 Overview

In this project, we will be working with past emergency data as well as geographical data so the need for a database is mandatory to keep large data manageable. There are two types of database that can be used, relational versus non-relational. With relational database, information are stored using keys and tables will be able to reference one another with the use of foreign keys. Non-relational database are used when dealing with huge data in order to make sure that relations do not interfere if there is a mismatch or errors with keys **?**. With this in mind, we decided to go forth and use a relational database to order our data and have an easy way to connect different tables together for a computer to search through. This method will allow the machine learning algorithm to have an easier time finding the data that it needs to make predictions.

3.2 Criteria

The technology options we considered for this category are ESRI ArcServer, Amazon Web Services, and MySQL. For this piece, we are looking for an easy to use platform in terms of accessing data and creating/inputting table values as well as will it be simple to incorporate it with our current coding frameworks (Python and R). We also want to know pricing to ensure that we are not overspending for storage for our database. Finally we want to know if each database service provide samples or free usage of their services before we buy in allowing us to test and see which platform suits our project.

3.3 Potential Choices

3.3.1 ESRI ArcServer

ESRI ArcServer is a geospatial repository allowing users to store the geograhical data with a portal for ArcGIS. It was developed for ArcGIS, allowing it to connect with a widely use geographic information system to have sophisticated rules and relationships be ready to use **?**. Its REST API makes it easy to call from programs and check up on the import data.

The flaw in this is that ESRI is mainly a geospatial repository, which means it was design for more geographical data to incorporate it with its ArcGIS program. ESRI also has an upfront cost and could range depending on the company. As of now, ESRI does not list their pricing on their website.

Pros: Uses REST API to access data, Attached to geographic software, API calls are made from Python, Java, JavaScript

Cons: Upfront cost, Mainly a geospatial repository

3.3.2 Amazon Web Services

The Amazon Web Services (AWS) is a cloud database storage service which supports relational databases, NoSQL and in-memory cache. They developed their own tools to manage the cloud data base such as Amazon DynamoDB, Amazon RDS, and Amazon Aurora. Each one of the tools listed has their own documentation on how to operate it as well as an explanation of the benefits of each tool. They also have Python integrated with their tools which allows for manipulation of the database from the Python application **?**.

Some things to keep in mind is the upfront cost that Amazon charges for these services. Each tool has their own specific pricing which means knowledge of what tools a developer will be using is vital to the cost of the services. Most of the pricing of each tool is leaning towards a price per hour usage. However, Amazon does offer free trial of some of the tools listed, allowing for testing and experimentation before settling in. Since the database is in the cloud, if AWS ever has an outage, operations will be hindered until services are restored.

Pros: Has Python integration, Variety of tools to setup database, Has some API calls, Free trials on some tools

Cons: Cost, Maintenance or Outages will hinder development.

3.3.3 MySQL

MySQL is an open-source relational database which has years of usage and development. Being open-source, there is a strong community that has developed different modules as well as extensive documentation on how to operate MySQL. There is a MySQL Python interface which makes development and integration of MySQL easier. It is free to

start up, allowing for development on personal machine much more easier and manageable. It includes a standard REST API which allows for easy access and manipulation of the data **?**

Even though it is free, there is an indirect cost of hosting the database if users decide not to use their Cloud base services. The pricing on the Cloud services is standing at $0.1815 per hour.

Pros: Free cost to start up, Has developed modules and documentation for ease of access, Python developed interface making development easier, Has REST API incorporated in cloud service

Cons: Cost of hosting database must be accounted for if using free version without plans for usage of Cloud service

3.4 Discussion/Comparison

Both MySQL and AWS provide free trials for their services allowing for users to test and generate conclusion on the type of platform they will need. They both also have a variety of tools with AWS providing a greater variety of specific tool for the correct user. ESRI ArcServer while promoting their servers for geospatial enthusiast, they do not offer any trials or testing. Users would need to buy in knowing exactly what they are getting from ESRI. Below is a table showing the options that was discussed and how they stand in our criteria.

Factors	ESRI ArcServer	AWS Database	MySQL
Price	✓	✓	✓
Framework Compatibility	X	✓	✓
Testing Availability	X	✓	✓

(Table 2): Showcases the comparison between the 3 different Database platform

3.5 Conclusion

With pricing in mind, it seems that AWS and MySQL seem to be tied as for being the main database platform we will be using. However, the edge has to be given to AWS for their variety in tools and software that is compatible with their platform. By allowing users to test their product before buying, we will be able to pinpoint the tools we want to use and know exactly what we are getting into.

4 Application Programming Interface

4.1 Overview

APIs are a set of protocols for building an application. They can be seen as a ”convenience” for many developers who needs a standard way to interface with an application. Instead of having to re-write code, developers can make API calls to request certain functions or information needed to advance their application. They are meant as tools to help stop developers from reinventing the wheel **?**. APIs will be used in this project to interface with our application to query the correct information from Code3 Visionary. Development of an API system would allow for other developers to use Code3 Visionary in their own application.

4.2 Criteria

The project will be mainly focusing on Web APIs and taking into consideration Simple Object Access Protocol (SOAP), Remote Procedure Call (RPC) and Representational State Transfer (REST) **?**. These classification must match our project model and what we hope to achieve in terms of developing and reliable API that can be used by other

developers. We want the API to allow users ease of access to the information they need in terms of readability of the information coming back as well as not giving too much (useless) information. The API must also be flexible and know the separation between public space and private. This is to allow internal developers to have access to application specific information that we might not necessarily want a normal user to have.

4.3 Potential Choices

4.3.1 SOAP

Simple Object Access Protocol (SOAP) relies on XML to describe the typed framework. Every parts of the operation from the XML structure of both the request and response is provided. When a request is made, a Web Service Description Language (WSDL) is used as a form of contract for the consumer and the service **?**.

Pros: Provides the right kind of information users are asking for, WSDL makes it easy to link up with an existing SOAP implementation

Cons: When changing something in the API, the WSDL must also change which means a new agreement must be made between the client and service, Gives more information then a user would need to know

4.3.2 RPC

Remote Procedure Call (RPC) can be describe the same as calling a function in JavaScript or Python. It takes in arguments and method names and uses JSON-RPC protocol to make the API calls. They are considered to be used for ”actions” having the procedure in their calls and a parameter to operate on those procedures **?**.

Pros: They are great for needing to do operations, provide a familiar interface to make calls

Cons: Not the right tool if user wants to retrieve information, might end up storing into the database if used incorrectly

4.3.3 REST

Representational State Transfer (REST) makes use of an already widely used language like HTTP and does not try to deviate from what is already considered ”standard”. It structures it requests/response into XML, YAML, and any other machine readable format like JSON. Objects are not strongly typed and meta data are structured hierarchically **?**. The REST API is also known as the most widely used API, being well used than SOAP and RPC.

Pros: Already using standardized form, Does not structure itself to one format, Allows for frequent updates without impact to consumers

Cons: Numerous amounts of different REST frameworks that is in the market

4.4 Discussion/Comparison

While REST and SOAP does provide user with the information that is request, SOAP does expose more underlining information that the user may not want to see. Even though RPC can retrieve information, it can also write information into the database if used incorrectly. Below is a table illustrating the 3 APIs and what they cover in our criteria.

Factors	SOAP	RPC	REST
Usability and Readability	X	✓	✓
Flexibility (Internal vs External)	✓	✓	✓

(Table 3): Showcases the comparison between the 3 different Web APIs

4.5 Conclusion

In regards of the API of choice, the one that we see having less issues to incorporate would be the REST API. By following a standard that everyone uses, and allowing for frequent updates to not interfere with the consumer, the REST API is easier to use than SOAP or RPC. Also a big bonus is that it supports different structures, which allows for increase in flexibility when data can be better represented in one form versus another. Finally by being popularly adopted, the REST API would have more documentation which means it would increase that ease of use for both us to develop and other consumers to use.

5 Review

To review the 3 technologies discussed in this paper were machine learning algorithms, databases, and APIs. Our of the three machine learning algorithms that was discussed, the best choice for the group is to use the Google TensorFlow. Although TensorFlow was developed with the C-language as its back end, this is better due to C being a low level language which allows it to run faster in the back end. TensorFlow also has great API support and Python integration for our project which makes it the best choice for what we are aiming for.

In terms of database platform, the most reasonable answer is to use AWS. AWS provides many tools that they develop themselves which helps developers create, maintain, and modify their databases. By allowing developers to also test out some of the tools for free first before having to buy in, gives a lot of flexibility for developers to find the right tools for the job. Lastly, since AWS is design with API integration, making calls to the database from our Python programs should be seamless.

With concern the the type of API we would want to use, the best one to consider would have to be the REST API. The REST API already builds upon a standardize framework that everyone knows (HTTP) which allows it to be easy to learn and understand. Instead of forcing developers to adhere to one type of structure, the REST API allows developers to choose what type of structure they see best fits their application. This would mean data will be represented in a clearer format and developers would not have to work around the structure to fix a problem. Finally, the wide adoption that the REST API already has allows it to be well know to the community and other developers. If a consumer is using are API, we would want the API to be something familiar to them to help with the ease of use.

By being able to take a look at the different options we have in each technology, we will be able to choose the correct option that would best fit this project. Knowing that there are other possibilities out there means that we can change our option if it does not work out in the long run. This paper helps illustrate the types of options that are available to us, and provides a breakdown of each option to show the best tools we can use to help us build our product. Consideration of all available options is always necessary when working on a big project. Not only for the ability to change if something occurs, but also the knowledge behind each pick that was decided in our project.