



College of Engineering

CS CAPSTONE PROBLEM STATEMENT
FALL TERM
OCTOBER 16, 2017

CODE3 VISIONARY

PREPARED FOR
LEVRUM DATA TECHNOLOGIES

CARL NIEDNER

Signature

Date

PREPARED BY
GROUP 3

KIEN TRAN

Signature

Date

BRIAN WILTSE

Signature

Date

Abstract

Code3 Visionary aims to use machine learning and artificial intelligence to predict need for emergency services. We will use data from various sources, such as prior emergency services call and dispatch logs to build predictive models for emergency event densities in a given time and place. These predictive models will be used to provide more input to Levrum's existing Code3 Strategist software, which allows users to build what-if scenarios based on various resource parameters.

CONTENTS

1	Definition	2
2	Proposed Solution	2
3	Metrics	3

1 DEFINITION

Levrum Data Technologies aims to optimize emergency services resource allocations. Levrum's existing software, Code3 Strategist, optimizes placement of emergency service resources based on what-if analyses. Users can adjust various parameters, such as response times, station locations, and workload to see the expected outcome of various emergency situations. Code3 Visionary is the next step in their software architecture.

This new project's goal is to use machine learning and artificial intelligence to predict need for emergency services. Machine learning algorithms will use data from prior events to create predictive models for what emergency service personnel can expect in a given time and place. Code3 Visionary will be used to enhance Code3 Strategist's capabilities, as it will inform the existing what-if scenario framework with predictions regarding what will likely occur, based on what happened in the past.

2 PROPOSED SOLUTION

Code3 Strategist has already had success with what-if scenarios, but Code3 Visionary's goal is to use data to anticipate densities of emergency service needs in a given time and place. We will begin by researching machine learning and artificial intelligence algorithms to best fit Code3 Visionary's particular optimization needs. One anticipated challenge is that there is a large amount of some types of data, such as emergency event time, location, and cause, but some sets of data we may be using will be sparse. Initial research will be aimed at the following questions:

- What algorithms will yield the greatest predictive power, given the amount of data we have of various types?
- What algorithms are best suited for this type of predictive model?
- What data is useful in making these predictions?
- What kind of data will be outputted to an end-user?

Although the proposed solution is subject to change depending on what research yields, after answering these questions, a few design decisions are reasonably certain. Our proposed solution involves:

- 1) implementing the algorithms with the highest yield in predictive power in a back-end calculation engine.
- 2) Training the algorithms with data from prior events.
- 3) Producing output that looks like an emergency dispatch call log
- 4) Implementing a REST API that will give Code3 Strategist access to Code3 Visionary's results.

We will need to implement a back-end calculation engine that will include algorithms for developing predictive models. We will concurrently need to build connectors to condition the data necessary for training these algorithms. The algorithms and connectors will be prioritized based on return-on-investment; that is, how much predictive power we can expect to gain.

Further, since the ultimate goal of the project is a set of machine learning algorithms that are accessible by Code3 Strategist, we will need to implement a fully encapsulated REST API. Code3 Strategist will then have access to the new predictive models via API calls.

Lastly, we will need to integrate the new functionality into Code3 Strategist. This will involve usability design decisions that are discussed further in the Metrics section.

Though a lot of design decisions will be informed by further research, the ultimate goal is a set of 'pluggable' machine learning algorithms that can be used by the existing Code3 Strategist to build more robust what-if scenarios.

3 METRICS

Metrics for success in this project fall into four categories:

- 1) **Predictive model.** We will constructed a trained model that will contain the necessary algorithms that will be train with prior data to generate predictions for future incidents. The model must be able to take in new data and used what it already knows from its training to create a perdictive output for the user. The predictions must have reasonable predictive power, and will be tested against actual data. Initial research will inform how much predictive power we can expect to be given the available data; however we will use time series data to get a measurement of baseline accuracy. A minimal metric is to implement algorithms that reliably achieve better than baseline.

The computational cost of the calculations must also be taken into account. Research must inform what the best-case scenario is for computational cost, and the user interface must be designed accordingly. User interface decisions are discussed further in item 4.
- 2) **REST API capabilities.** We will have a functioning API that will be callable from Code3 Strategist. We will design a set of API calls that will allow Code3 Strategist to send and receive data from the calculation engine. The REST API must, at a minimum, be able to get results from the predictive algorithms we implement.
- 3) **Data.** Data being input into machine learning algorithms must be conditioned for the calculation engine. Different databases, even those holding similar data, will likely be configured with different schemas, and we will certainly be dealing with various types of data. We must determine what data will yield the greatest predictive power, prioritize those data, and implement modules to condition that data.
- 4) **End-to-end usability.** Code3 Strategist needs to be able to use data generated in the calculation engine. Code3 Strategist's code base must be updated to take in the predictive model produced by Code3 Visionary. Moreover, usability must be taken into consideration. As mentioned in metric 2, optimization calculations are likely to be expensive. If the predictive power of the algorithms is substantially better than baseline, users will likely tolerate waiting some time for a solution. However, if the user is presented with myriad options, each of which would result in refreshing the calculation for updated output, we must take care to not test the user's patience. If expensive, repeated calculations are inevitable, we must properly nest calculations in order to ensure that only downstream calculations are refreshed.

Ultimately, at the end of the project, a user should be able to use predicted density of incidents as an additional parameter to its functionality for creating what-if scenarios.