**Applied Artificial Intelligence and Machine Learning in Retail Sales and Order Forecasting**

**Software Development Plan**

**Version 1.0**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 08/OCT/2019 | 1.0 | Initial Release | Shahzeb Khalid |
| 09/OCT/2019 | 1.1 | Structural Changes | David Kelchner |
| 10/OCT/2019 | 1.3 | Expanded some sections, modified others | Edward Coombes |
| 22/OCT/2019 | 1.4 | Added additional sections to the document | Edward Coombes |

**Table of Contents**

1. Introduction 5

1.1 Purpose 5

1.2 Scope 5

1.3 Definitions, Acronyms and Abbreviations 5

1.4 References 5

1.5 Overview 6

2. Project Overview 6

2.1 Project purpose, scope and objectives 6

2.2 Assumptions and constraints 6

2.3 Project deliverables 6

2.4 Evolution of the Software Development Plan 6

3. Project Organization 6

3.1 Organizational Structure 6

3.2 External Interfaces 6

3.3 Roles and Responsibilities 6

4. Management Process 6

4.1 Project Estimates 6

4.2 Project Plan 7

4.2.1 Phase Plan 7

4.2.2 Iteration Objectives 7

4.2.3 Releases 7

4.2.4 Project Schedule 7

4.2.5 Project Resourcing 7

4.2.6 Budget 7

4.3 Iteration Plans 7

4.4 Project Monitoring and control 7

4.4.1 Requirements management plan 7

4.4.2 Schedule control plan 7

4.4.3 Budget control plan 8

4.4.4 Quality control plan 8

4.4.5 Reporting Plan 8

4.4.6 Measurement Plan 8

4.5 Risk Management plan 8

4.6 Close-out plan 8

5. Technical process plans 8

5.1 Development Case 8

5.2 Methods, tools and techniques 8

5.3 Infrastructure plan 8

5.4 Product acceptance plan 8

6. Supporting process plans 8

6.1 Configuration management plan 8

6.2 Evaluation plan 9

6.3 Documentation plan 9

6.4 Quality assurance plan 9

6.5 Problem resolution plan 9

6.6 Subcontractor management plan 9

6.7 Process improvement plan 9

7. Additional plans 9

8. Annexes 9

9. Index 9

**Software Development Plan**

# **Introduction**

This document serves to provide a necessary overview of the project, and outlines our plan for bringing this project to completion. The goal of this project is to accurately predict retail sales in order to create product orders which will satisfy the needs of the store, without overstocking on any products. Our software will have three main features: applied machine learning for order forecasting, the generation of product orders in line with the input of the commercial product ordering system PDI, and also an error trapping subsystem to alert the IT (Information Technology) staff should an error occur so that it may be properly addressed. This document will describe assumptions, constraints, and deliverables of this project while also addressing the management of the software development process from a time, technical, and financial perspective.

## **Purpose**

The purpose of this project is to forecast future sales at individual cumberland farms stores, for the purposes of generating accurate product orders. This will mitigate the financial losses that come from problems such as overstocking and understocking of goods.

## **Scope**

* Evaluation of historical sales data at each store.
* Consumption and consideration of local environment data specific to each store.
* Forecast retail sales for each store one week in advance.
* Generate product orders which encompass the forecasted sales.
* Integration with existing warehouse and product ordering system PDI.
* An alert subsystem which will notify IT staff about potential errors that occur with the algorithm.

## **Definitions, Acronyms and Abbreviations**

* **Artificial Intelligence** - The ability of computers to learn and perform tasks in a smart way, without being explicitly given exact instructions.
* **Classification** – A subset of machine learning problems where the dependent variable is finite and discrete.
* **Clustering** – A technique that groups data points together in such a way so that all data points in a single group have similar features, and exist close to each other in high dimensional space.
* **Data** - information that represents values attributed to parameters which has been translated into a form that is efficient for movement or processing by a human or a machine.
* **Error Trapping –** The process of catching errors in parts of a program where it is likely to occur and making some kind of alert and log of the occurrence of the error.
* **Meta-Predictors** - An independent variable in a dataset to be used in machine learning, which is itself the output of a machine learning algorithm.
* **Supervised Learning** - Machine learning methods which train on data where the dependent variable is labeled.
* **Unsupervised Learning –** Machine learning methods which train on unlabeled data.

## **References**

1. “Vision Document”, 9/26/19, Project 4 Team
2. “Glossary”, 10/08/19, Project 4 Team
3. “Software Requirement Specification”, 10/15/19, Project 4 Team
4. “Iteration Plan”, 10/17/19, Project 4 Team
5. “GitLab Basics Guide”, No Date Provided, GitLab
6. “Machine Learning Models For Sales Time Series Forecasting”, 01/18/19, MDPI
7. “How to Predict A Time Series”, 07/08/19, Towards Data Science
8. “A Beginners Guide to LSTMs and RNNs”, No Date Provided, Skymind
9. “Time Series Prediction”, 11/13/18, Siraj Raval

## **Overview**

The rest of this document describes the project organization, and plans for the management process, technical process and supporting processes for the project. This will encompass the organizational structure, external interfaces, budget, schedule, infrastructure and evaluation of the project. Each item will be defined and described in it’s own section.

# **Project Overview**

## **Project purpose, scope and objectives**

The object is to improve capital utilization and customer experience by eliminating out of stock and overstock situations in retail stores and warehouse. Predictions should be made one week prior, and the system should be able to send a text message and/or an email regarding the matter of resupplying the product. The end goal is to develop an algorithm that is capable of accurately stocking the store by anticipating future needs.

## **Assumptions and constraints**

The primary assumption of this project is the proper supplementation of data. The project does not implement any form of web scraping to gather environment data (weather, local events, traffic, etc). However the project does have the capability to perform requests from the cumberland farms database. The project also has multiple software dependencies including Tensorflow, Keras, Scikit-learn, Scipy, Pandas, and Twillio. There is an implicit assumption that these tools are installed when the software begins to run.

The primary constraint of this project is the schedule. Developing this system within the allotted time frame will require good time management. A secondary constraint of this project is budget; in order to send text message alerts, and to run algorithms which handle vast quantities of data money is required to pay for access to the Twillio API and to rent a server which meets the performance requirements of the algorithm.

## **Project deliverables**

The following is a list of deliverables as described by the client:

* 11/15 - An algorithm which consumes historical sales data, and environment data.
* 11/15 - A Sales performance Report, describing actual vs. forecasted sales by store.
* 11/15 - A solution which generates product orders by store.
* 10/20 - Error Trapping Code, which sends alerts via email and sms
* 11/10 - Supporting Documentation
* 10/25 - Process Flow Diagrams
* 11/15 - Source Code with in-line documentation.
* 10/25 - Database Schema
* 11/15 - Data Dictionary

## **Evolution of the Software Development Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| Item Name | Revision | Criteria | Date |
| Project | 1.0 | Software which implements alert subsystem, and consumes historical sales data to make predictions using machine learning methods and generates orders based on the prediction. | 11/15/2019 |
| Software Development Plan(SDP) | 2.0 | The SDP will be revised to respond to client feedback after the Minimum Viable Product is reviewed with the client. | 11/20/2019 |
| Project | 2.0 | An updated version of the software which consumes local environment data, as well as historical sales data. Also responds to any additional or changed requirements after the delivery of the Minimum Viable Product. | May 2019 |

# **Project Organization**

## **Organizational Structure**

CTO of Cumberland Farms Sorin Hilgen is the main client for the project along with Ian St.Pierre, Dan Champagne and Charles Jarrett. Some other important stakeholders in the project include Prof. Valova and Ian Eichorn. The project team consists of the Project manager, Edward Coombes, Scrum master Owen Velho, software developer Shahzeb Khalid, software developer David Kelchner, and software developer Irene Sun.

The team will be divided into two sub-teams for the purpose of most sprints. The first sub-team will be focused on developing and improving the machine learning algorithm for use in the software. The second sub-team will be focused on the documentation deliverables, as well as the alert subsystem. In this manner it is possible for the team to focus on two things at once, while still maintaining a steady pace of improvement on the machine learning model. The first sub-team will consist of Edward Coombes, Shahzeb Khalid, and David Kelchner. While the second sub-team will consist of Owen Velho and Irene Sun. In this way the leadership is divided amongst the sub-teams while allowing each person to perform tasks consistent with their skills.

## **External Interfaces**

This project will interface with the Cumberland Farms team overseeing this project.

Cumberland Farms:

* Dan Champagne
* Ian St.Pierre
* Charles Jarrett
* Sorin Hilgen

## **Roles and Responsibilities**

|  |  |
| --- | --- |
| **Person** | **Process Role** |
| Edward Coombes, Senior Manager | Project Manager  Deployment Manager  Requirements Reviewer  Architecture Reviewer  Configuration Manager  Change Control Manager |
| Owen Velho, Software Developer | Management Reviewer  Requirements Reviewer  SCRUM Master  Test Manager  Test Analyst |
| David Kelchner, Software Engineer  Shahzeb Khalid, Software Engineer  Irene Sun, Software Engineer | Designer Implementer Code Reviewer Integrator Test Designer Tester Technical Writer |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Activities** | **Document name** | **Elaboration** | **Technical Control** | **Approval** |
| Project management | Project Management Plan | Owen Velho | Owen Velho | Edward Coombes |
| Setting up the Development tools | Software development plan | David Kelchner, Shahzeb Khalid | Edward Coombes | Edward Coombes |
| Software specifications | SRS | Owen Velho,  Irene Sun | Edward Coombes | Edward Coombes |

# **Management Process**

## **Project Estimates**

This project has two sources of financial expense: the sever to run the algorithm, and the API to send text messages.

The server is priced at $0.229/hour for pay as you go, or $0.1236 for a whole year. To purchase this using pay as you go style pricing until the end of the academic year would cost $1188. To purchase this for a whole year would cost $1056.

The text messaging service that we will be using (Twillio) costs about $0.075 per message. Assuming we send 500 messages over the course of the project we will spend an estimated $37.5 on text messaging. The service also has a monthly fee of $1, assuming it is purchased until the end of the academic year that is another $8 for a total of $45.5 estimated cost from text messaging.

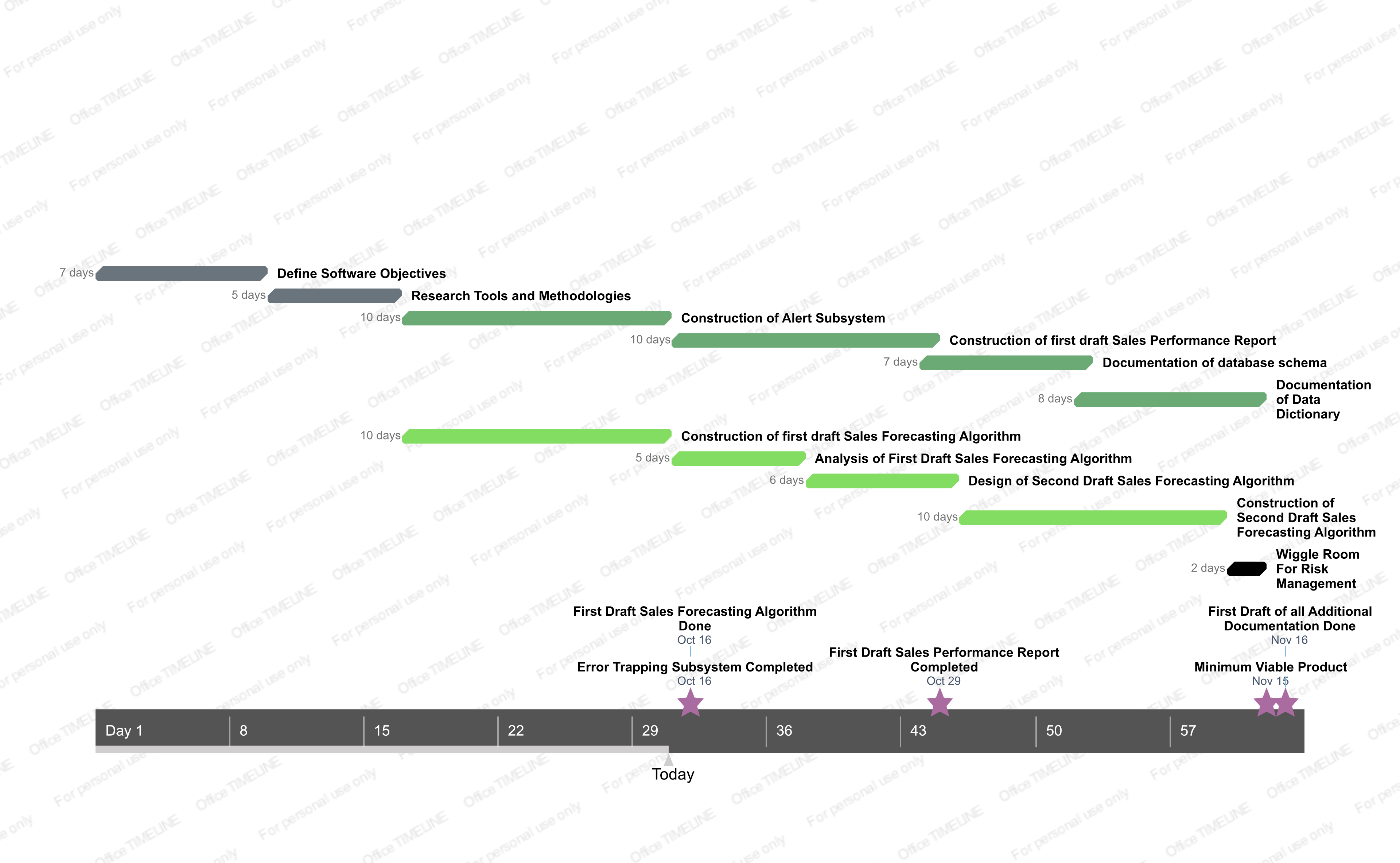
Overall, the current estimated budget for this project is $1101.50. This budget will be revisited and reestimated after November 15th, when the minimum viable product is due. The budget may need to be adjusted to fit future data needs, as well as to account for any additional APIs which may be needed to implement potential additional services as specified by the client.

The project is on a tight schedule to be done by the end of the academic year. However we estimate that the minimum viable product due date of November 15th will be when our first full release will occur. After that we will hope to have production ready updated releases every other week until the end of the academic year; with each release improving in accuracy, or improving our understanding of how to make the model more accurate.

## **Project Plan**

### Phase Plan

In section 3.1 two separate sub-teams were defined. These sub-teams will mostly work on separate tasks, while working in tandem to achieve the goals defined by the client. See below for a matrix depicting a generalized task breakdown, as well as major milestones from the beginning of the semester to the due date for the minimum viable product. This will be revised as needed to enable this project to work dynamically.



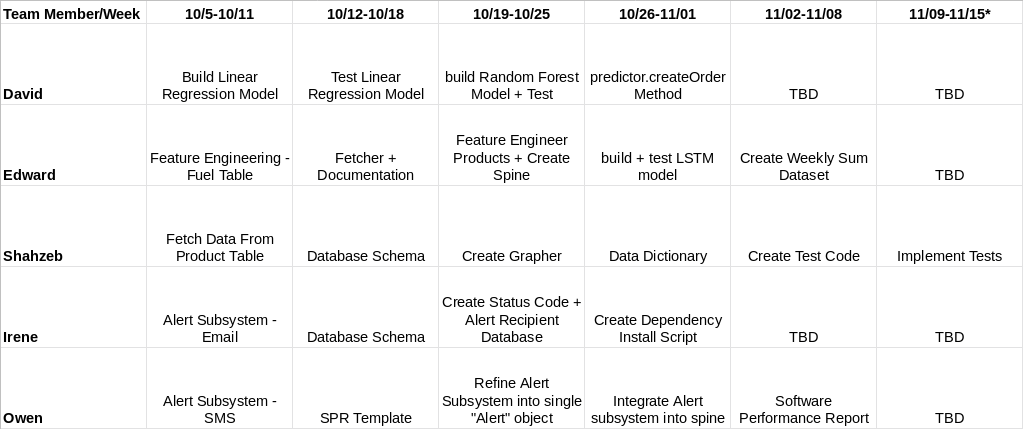
### Iteration Objectives

During each weekly scrum iteration each member of the team will be assigned a task to move us along in our goals. See below for a task matrix describing whateach person is hoping to accomplish during each weekly

### Releases

* Release 1 – Beta
  + Released on 11/02
  + Will implement machine learning algorithm to predict future sales using historic sales data. Alert subsystem will be fully constructed.
* Release 2 – Minimum Viable Product
  + Released on 11/15
  + Will implement machine learning algorithm to predict future sales using historic sales data in both a production and a learning mode. Alert subsystem and contacts database will be fully set up to send alerts should any error occur, and when the system finishes running.
* Release 3 – Version 2.0
  + Released on 12/09
  + Will improve on the algorithm to generate a skeleton Sales Performance Report from the data.

### Project Schedule



### Project Resourcing

#### Staffing Plan

This project requires 5 total staff members. One must serve as a sort of executive leader of the software development team, and another must serve as a scrum master who helps the team leader be agile and implement the scrum methodology. Other team members will require skills in either machine learning specifically, or more generalized experience working with data in a database environment. Strong Python skills, and comfort in a terminal environment are also very useful. The ideal software developer will also have good interpersonal skills, which will enable them to work effectively in a team environment.

#### Resource Acquisition Plan

After reviewing all of the resumes submitted, emails were sent out to all members of the UMass Dartmouth CIS Class of 2020 who matched the description outlined in section 4.2.5.1. After that, responses were measured and two strong candidates were chosen. This allowed for the team to have a strong base going into the job fair hiring process. During the job fair two candidates were selected based on their skills, but also the potential they were perceived to have to work well with the group as a whole.

This ensured that the team had no only strong base for technical skills, but also a strong base for effective interpersonal communication and cooperation.

#### Training Plan

Team members working on the sales forecasting algorithm will need a basic understanding of machine learning to contribute meaningfully to the construction of the sales forecasting algorithm. This training should be completed by 10/02/2019. This training should provide the trainees with an understanding of machine learning, and knowledge about various methods used for regression, classification, clustering, and dimensionality reduction.

### Budget Plan

This project has two sources of financial expense: the sever to run the algorithm, and the API to send text messages.

The server is priced at $0.229/hour for pay as you go, or $0.1236 for a whole year. To purchase this using pay as you go style pricing until the end of the academic year would cost $1188. To purchase this for a whole year would cost $1056.

The text messaging service that we will be using (Twillio) costs about $0.075 per message. Assuming we send 500 messages over the course of the project we will spend an estimated $37.5 on text messaging. The service also has a monthly fee of $1, assuming it is purchased until the end of the academic year that is another $8 for a total of $45.5 estimated cost from text messaging.

Overall, the current estimated budget for this project is $1101.50. This budget will be revisited and reestimated after November 15th, when the minimum viable product is due. The budget may need to be adjusted to fit future data needs, as well as to account for any additional APIs which may be needed to implement potential additional services as specified by the client.

## **Iteration Plan**

Refer to the Iteration Plan contained in the references section of this document.

## **Project Monitoring and control**

### Requirements Management Plan

The team lead will maintain a list of requirements that are necessary for the successful completion of this project as outlined above. Upon the completion of key components of the project, and the production of deliverable items will be checked off of this list so as to maintain an up to date list of requirements that need to be completed.

### Schedule Control Plan

Progress will be monitored in Trello and by Scrum meetings carried out 3 times a week. If there is any deviation from the planned schedule, notes will be taken on what problems occurred and any roadblocks. These will be addressed by the SCRUM master and Project manager to help get the project back on track.

### Budget Control Plan

As outlined in the budget plan above, this project has a fixed cost and should at no point in time require additional funding to move forward. Should a situation arise where the requirements are changed, and additional funding is required to add additional functionality to this project then the budget will be revisited. If the team approaches the monetary limit of the project, then an analysis will be done on the importance of each item on the budget, and the availability of free alternatives to that item.

### Quality Control Plan

After the development of the beta model, a code review will be performed. Each member will review the code generated by another member of the team. Each reviewer will “grade” the code based on the following criteria: ease of understanding, explanatory in-line documentation, proper formatting, “compactness” (if the code accomplishes everything possible in the minimum amount of lines possible without obfuscating the actions of the code).

Should a reviewer be completely unable to understand the code, they will put in-line documentation of the questions that they have. Then the reviewee will sit down with the reviewer and explain what is happening while also adding additional in-line documentation describing the parts of the code which confused the reviewer. In this way we can enhance the understanding team members have of different components of the code; while also providing in-line documentation which will aide non-team members in understanding the functionality of the code, and completing a requirement of the client that all code has in-line documentation.

### Reporting Plan

The team lead will write a weekly report describing the accomplishments, goals, and issues encountered by the team. The scrum master will also write a report on a bi-weekly basis describing how well the team adhering to the principals of scrum and staying on task and on schedule. In addition to this, the team has a bi-weekly meeting scheduled with the client to discuss project deliverable progress.

As another reporting measure that is outlined in the project description of the client, a sales performance report will also be created. A full sales performance report only needs to be created once, in order to analyze the performance of our algorithm as well as the performance of the various cumberland farms stores. It is our hope that in addition to this one-off sales performance report, that the tools that we are creating for ourselves to analyze the performance of the algorithm can also be utilized to create a “skeleton” of a sales performance report on production runs of the software. This could be used by the marketing department to analyze the performance of the stores as well as the performance of the machine learning algorithm.

### Measurement Plan

Details on the statistical methods that will be used to measure the performance of the machine learning algorithm are contained in the Vision Document.

## **Risk Management plan**

Risk is assessed on an as needed basis. Much of the risk associated with the project is man hours and scheduling. We will:

* Keep all members regularly updated on the status of each part
* Give a detailed schedule of events that will occur during the curriculum
* Every team member gives a list of times and dates they are available
* The client is notified and asked when they are available
* Tasks are divided up based on how much time the member has to work on it

## **Close-out plan**

In order to close out the project, and give all deliverables to the client the team will use a gitlab account to store the working iterations of the code. All documentation will be included on this gitlab as well. As a secondary measure, all of the code will be available on the server rented by the client so that any and all data as well as the source code is accessible by the client. As a tertiary measure, the deliverable documents and the source code for the project will be placed on a flash drive and hand delivered to the client at CIS day.

# **Technical process plans**

## **Methods, tools and techniques**

## Methods and tools are thoroughly discussed in the Software Requirements Specification Document [3].

Techniques are subject to change, but at this point in time the running idea is to calculate for each product, the net sale from the previous three days as a feature called “lag”. The difference between each day will also be calculated as a feature called “delta”. To encode time of year data seasonality must also be computed. In order to be more specific than just “Fall, Winter, Spring, Summer” this seasonality is captured by using the dates of astrological signs (Aries, Virgo, Capricorn, etc). The hope is that incorporating the recent change to the net sale of a product, as well as the time of year trends can be established and a regression algorithm can accurately predict the future net sale values [6][7][8][9].

## **Software Infrastructure Plan**

Refer to the Software Architecture Document.

## **Product Acceptance Plan**

See section 2.3 for a list of product deliverables. The product will be accepted as long as all of the deliverables are created, and the algorithm works accurately within the timeframe of a single day.

# **Supporting process plans**

## **Configuration management plan**

Configuration management will be handled by GitLab. Instructions for using GitLab are available on the GitLab website, on the “GitLab Basics” page. Deliverable documentation will also be maintained on the GitLab. This is to ensure that the client has easy access to all the relevant data of the project.

## **Evaluation plan**

The project will be evaluated using the following criteria: accuracy ratings of the model as defined by the Vision Document [1], as well as successful completion of all deliverables as defined in section 2.3 of this document. Evaluation metrics will be measured and recorded in the Sales Performance Report to be delivered to the client.

## **Documentation plan**

All software written will have extensive in-line documentation explaining the implementation details of each module. In addition to this, the team has created many diagrams explaining on a more general level how the software operates. These diagrams will all be compiled into a manual that will be updated whenever a structural change is made to the program. This manual will be delivered to the client at the completion of the project.

## **Quality Assurance plan**

Please refer to section 4.4.4 for the Quality Control Plan.

## **Problem Resolution plan**

In the event that a team member encounters a technical problem, they will bring it up to the group. In this way, we can all brainstorm solutions for the team member to try. A change in perspective, and a fresh set of eyes often solves most problems. Should that team member still be unable to solve the problem, the task will either be pushed into the backlog or reassigned to another member of the team. In this way the team is ensured to not be slowed down by one single issue. After an issue has to be reassigned, the team lead and the scrum master will work together to reassign it to the person with a skill set that makes them the most likely to be successful in completing the problem task.