Fidelity Procurement Categorization of the Current Spend Trends

Project Team:

Luke Deratzou lderatzou@wpi.edu

Neville Ingram nlingram@wpi.edu

Olivia Reneson owreneson@wpi.edu

Ziqian Zeng zzeng@wpi.edu

Project Advisor

Professor Robert Sarnie

School of Business

Project Advisor

Professor Wilson Wong

Department of Computer Science

Project Advisor

Professor Marcel Blais

Department of Mathematical Sciences

Abstract

Fidelity Investments utilizes spend procurement software to make a multitude of financial decisions. Before the data can be available within the software, the data needs to be classified. Fidelity uses machine learning models to classify spend data and would like the models to be more accurate and efficient to leverage spend data. Miscategorized spending distorts documentation and reduces the value of the spend data. To enhance the classification process, Fidelity has asked our team to improve the current models by using Python multithreading on Amazon EMR and introducing models with lower run time and higher accuracy. Our team met Fidelity's needs by introducing two new, faster models (K Nearest Neighbors and Decision Trees) as well as fine-tuning pre-existing models to increase accuracy.

Acknowledgments

We would like to begin by thanking everyone we had the pleasure of working with throughout the term. Thank you to our advisors: Professor Sarnie, Professor Wong, and Professor Blais for their guidance and assistance throughout this project.

We would also like to thank our contacts at Fidelity: Michael Zaharis, Vivianne Luk, Kiran Venkatadusumelli, Anuj Jain, and Lakshmi Devi Sopaku. Everyone at Fidelity was very helpful and played crucial roles in aiding our team to complete this project.

Executive Summary

Fidelity Investments is a financial service corporation that supports a wide range of products and businesses. To maintain these businesses, Fidelity must remain organized and efficient. A product used to help attain this is their procurement software. Within the software, users can find data from all purchases made within Fidelity. This data includes historical spend data, the vendor catalog, and external vendors. The user might seek this information to leverage spend data, find a vendor, or make price estimates. Properly documented data is essential for calculating spending within sectors and leveraging spend data for discounted prices.

To classify the data before entering it into the procurement software, Fidelity has two machine learning models. Fidelity's current problem is that the run time and accuracy of the models are not up to their standards. For the models to be more effective, Fidelity would like them to be as accurate and fast as feasibly possible. To accomplish this, Fidelity has asked our team to improve the performance of their current models as well as research and implement new machine learning models.

Within the project term, the team was able to improve the performance of Fidelity's current machine learning models by optimizing the models and multithreading on Amazon EMR. We also researched three new models, and from those we choose two to recommend to Fidelity. The first of these models is K Nearest Neighbors, the second being decision trees. With these models, the team received similar accuracy rates to pre-existing models but greatly reduced the run time.

To maintain organization within our team, we employed Agile Scrum Methodology. We found this to be immensely helpful, especially when the team encountered blocks to our progress. Gaining access to necessary software was an issue the team faced throughout the term. When we were unable to continue a user story because of software access it became important that we document the issue and find ways to continue work. Doing so resulted in the team maintaining a steady workflow.

Table of Contents

Abstract **ii**
Acknowledgments **iii**
Executive Summary **iv**
Table of Contents **v**
Table of Figures **vii**
Table of Tables **viii**
1. Introduction **1**
1.1 Fidelity 1
1.2 Problem Description 1
1.3 Goals 1
1.4 Deliverables 2
2. Research **4**
2.1 Spend Data Classification 4
2.2 Machine Learning 5
2.2.1 Random Forests 5
2.2.2 Linear Support Vector Classifier 6
2.2.3 Decision Trees 7
2.2.4 K Nearest Neighbors 8
2.2.5 Multilayer Perceptron 9
2.3 Current Framework 10
2.4 Project Benefits 11
3. Methodology **12**
3.1 What is Agile Scrum? 12
3.1.1 Workflow of Agile Scrum 12
3.1.2 Daily Scrum 13
3.2 Sprints 13
3.2.1 Sprint Planning 14
3.2.2 Sprint Review 14
3.2.3 Sprint Retrospectives 14
3.3 Scikit-learn 14
4. Software Development Environment **16**
4.1 Project Management Software 16
4.1.1 Discord & Microsoft Teams 16
4.1.2 JIRA (Fidelity Company Portal & Team Portal) 16
4.2 Programming Environment 16
4.2.1 Visual Studio Code 1.57.1 16
4.2.2 Python 3.6.4 (Anaconda3 5.1.0) 17
4.3 Software Tools 17
4.3.1 BMC Control-M 9.0.20.1000 17

4.3.2 WinSCP 5.9.6 17
4.4 Data Sources 18
5. Software Requirements 19
5.1 Software Requirement Gathering Strategy 19
5.2 Functional and Non-Functional Requirements 19
5.3 Epics and User Stories 20
6. Software Design 24
6.1 System Architecture 24
6.2 Snowflake Database Usage 25
6.3 High-Level Development Workflow 26
6.4 Machine Learning Model Investigation Process 27
7. Software Development 29
7.1 Sprint 1: 10/25 - 10/30 29
7.2 Sprint 2: 10/31 - 11/6 31
7.3 Sprint 3: 11/7 - 11/13 34
7.4 Sprint 4: 11/14 - 11/20 38
7.5 Sprint 5: 11/21 - 11/27 41
7.6 Sprint 6: 11/28 - 12/4 43
7.7 Sprint 7: 12/5 - 12/11 48
7.8 Product Burndown Chart 52
8. Business and Project Risk Management 54
8.1 Risk VS Reward 54
8.2 Risk Culture 55
8.3 Additional Risks 56
8.3.1 Operational Risks 56
8.3.2 Financial Risks 57
8.3.3 Reputational Risks 57
8.3.4 Innovation and Change Management Risks 58
8.3.5 Training Risks 58
9. Assessment 60
9.1 Business Learnings 60
9.2 Technical Learnings 62
9.3 Takeaways 63
10. Future Work 64
11. Conclusion 66
12. References 69

Table of Figures

Figure 2.1 - Bagging Algorithm 6
Figure 2.2 - Linear Support Vector Classifier Maximization Problem 7
Figure 2.3 - Residual Sum of Squares Equation 8
Figure 2.4 - Conditional Probability equation for class labels in KNN 9
Figure 2.5 - Example Multi-Layer Perceptron 10
Figure 3.1 - Agile Scrum Workflow 13
Figure 6.1 - System Architecture Diagram (Corporate Technology Group, 2021) 25
Figure 6.2 - High-Level UML Activity Diagram of Development Workflow 26
Figure 6.3 - ML Model Investigation Process Diagram 27
Figure 7.1 - Product Burndown Chart 52

Table of Tables

Table 7.1.1 - User Stories in Sprint 1 30
Table 7.1.2 - Total Story Points in Sprint 1 30
Table 7.2.1 - User Stories in Sprint 2 31
Table 7.2.2 - Total Story Points in Sprint 2 34
Table 7.3.1 - User Stories in Sprint 3 37
Table 7.3.2 - Total Story Points in Sprint 3 37
Table 7.4.1 - User Stories in Sprint 4 41
Table 7.4.2 - Total Story Points in Sprint 4 41
Table 7.5.1 - User Stories in Sprint 5 43
Table 7.5.2 - Total Story Points in Sprint 5 43
Table 7.6.1 - User Stories in Sprint 6 47
Table 7.6.2 - Total Story Points in Sprint 6 47
Table 7.7.1 - User Stories in Sprint 7 50
Table 7.7.2 - Total Story Points in Sprint 7 50
Table 11.1 - Summarized ML Model Performance on 5 Million Rows of Spend Data 67

1. Introduction

1.1 Fidelity

Fidelity Investments is a large financial services corporation that supports a diverse range of customers. To provide to each customer segment, Fidelity has created several branches that cater to specific needs. This vast and complicated business structure makes it important that Fidelity maintains internal organization and efficient practices. Fintech, or finance technology, is the most effective way for Fidelity to accomplish this (Fidelity, n.d.).

1.2 Problem Description

Spend categorization is a subject that should not be overlooked when considering the internal organization of Fidelity. The ability to properly categorize the amount spent within each of Fidelity's sectors is instrumental to the business's success as this data documents yearly spending within the sectors. Fidelity currently classifies spending using two different types of machine learning models, a linear support vector classifier (SVC) and a random forest classifier (RFC). These methods are time-consuming: the Linear SVC model takes about 12-15 hours to complete training and the RFC model takes about 8-9 hours to complete training. In the case that spending is incorrectly classified by either of these models, Fidelity also has a spend classification correction form. For better use, the correction form requires an improved user interface and added features. Users are currently only able to redefine level 3 commodity, the highest level of spend classification. If the incorrect classification is caused by other criteria, there is no way for the user to edit the criteria and correct the classification. To make the classification form more beneficial, features need to be added that will allow users to also edit selection criteria.

1.3 Goals

The team has been asked to improve the performance of the Linear SVC model and RFC model using Python multithreading on Amazon EMR, as well as introduce new machine learning

models that yield higher accuracy along with lower run times. We researched three new machine learning models and implemented two of them. Alongside the models, we researched the benefits of using natural language processing and multilayer perceptrons. Once implemented, we compared the accuracies and time to run then picked the top-performing models. The final models were included in a guide for future use and presented to Fidelity.

1.4 Deliverables

The team was able to successfully make improvements to Fidelity's current machine learning models as well as introduce two new models. To improve the performance of existing models, the team began by optimizing the hyperparameters to increase the accuracy. As a result, the accuracy rate of random forests increased from 94% to 99%. Linear SVC's accuracy rate increased from 93% to 99%. During this process, we learned that the pre-existing code had already been multithreaded. In addition to working with the pre-existing models, the team also researched k-nearest neighbors (KNN), decision trees, and Gaussian process as potential models to implement. We discovered early on that the Gaussian process would not be a successful model because of the long run time and decided to expand our research to natural language processing (NLP) and multilayer perceptrons (MLP). It was with the implementation of NLP into our machine learning models that the team learned that NLP was beneficial to decision trees but not KNN. Therefore, the team decided to pursue KNN and decision trees without NLP as our models to recommend. To further reduce the run time of our chosen models, the team tried to implement Python multithreading into KNN and decision trees. We concluded that multithreading with scikit-learn is not feasible, as multiple threads cannot be handled. In our testing, all models came back with a lower accuracy when multithreaded. Instead, we continued with single-threaded models and multiprocessing where applicable. Alongside our models, we documented our workflow, required infrastructure with how to gain access, and new AI/ML model scripts for Fidelity's future use.

The team was not able to pursue the second task (adding features to the classification correction form) within the project term because of progress blockers within the first task. The two main blocks were confusion surrounding the pre-existing state of the project and gaining

access to necessary software. We were given access to documentation of a previous intern's work on the project, but this did not align with what we had been asked to do. The team and our manager also had trouble discovering who had worked with this intern to ask them questions about the state of the project. Through reading the code and several meetings with individuals who had some familiarity with previous work, we were able to piece together past progress and where we needed to continue work. This was time-consuming, resulting in the team deciding to only focus on the first task. The second task has been documented as future work within the team’s project guide for Fidelity.

2. Research

2.1 Spend Data Classification

Spend data classification is the organization of spend data into similar, previously defined categories (Accelerated Insight, n.d.). These categories often contain subcategories to further define the data. Once organized, spend analytics provides more intelligent insight into how the provided data affects the company. To provide useful insights, the classification engine must be efficient and accurate. Classification engines that take too many resources or do not properly represent actual spending will not provide the company any benefit.

In 2019, Emir Ombasic, a data scientist at JAGGAER, presented automatic spend classification as a potential solution for inaccurate spend classification. Natural language processing techniques are used to process a description or product number and provide the European standard ECLASS product classification or US UNSPSC product classification (NewsBank, 2019). The JAGGAER automatic spend classification model utilizes a procedure that can be applied across many spending categories because it only requires a text description or product number. JAGGAER scientists have been able to implement the classification model to help their customers analyze their spending and make more informed decisions (NewsBank, 2019).

Through the Linear SVC and RFC models, Fidelity is forming an automatic spend classification model fit for their purposes. This model is intended to help Fidelity improve the organization of their current spend data and create better predictions for the future based on said data. Automatic spend classification models have many benefits for large companies such as Fidelity, as mentioned by Ombasic, “Once companies are able to match articles to categories, it is much easier to check the spend per category and spot increasing procurement costs. This is especially true for large corporations where the orders are made in large volumes and range from stationery up to laboratory equipment and specialized parts” (NewsBank, 2019). To make current models better fit Fidelity’s needs, reducing the time to run the models and improving the user interface of the correction form are essential.

2.2 Machine Learning

Machine learning utilizes artificial intelligence to allow systems to learn and adapt without being given direct instruction. The learning is done by finding patterns or examples in provided data, which the algorithm is then able to apply in the future to make decisions (Expert.ai, 2020). Past work on Fidelity's framework for classifying spend data has also included natural language processing (NLP). NLP uses computational linguistics in combination with statistical, machine learning, and deep learning models to enable systems' understanding of human language. With NLP, computers can process human languages through text or voice data. Within Fidelity's system, NLP is used to read inputted spend data and classify it into one of many categories (IBM, 2020).

Future work accomplished by our team implements multilayer perceptrons (MLP). MLP is modeled after a biological brain on a simpler level and is often used for predictive tasks. As the name suggests, MLP uses a multi-layered structure to "pick out (learn to represent) features at different scales or resolutions and combine them into higher-order features." (Brownlee, 2016) This structure allows MLP to take in training data and learn how to relate it to an output variable. Once this is done, the model can be used to make predictions, such as predicting the classification for spend data.

2.2.1 Random Forests

Random Forests is a supervised machine learning algorithm that grows and combines multiple decision trees to create a "forest". The multiple uncorrelated, individual decision trees each give a classification and the final classification is the one with the most overall votes from the set of decision trees (forest). In order to separate the data for each of the decision trees, the model is created using an ensemble method. An ensemble method combines predictions from multiple machine learning algorithms to make more accurate predictions than a single model. The ensemble method used is called Bootstrap Aggregation, also known as Bagging. Bootstrapping is when you choose random samples from a set with replacement. Bagging aims to reduce the variance introduced in decision trees and it does this by averaging the set of data

points (observations). Specifically, given a set of n independent observations Z1,...,Zn, each with variance σ^2, the variance of the mean $\bar{Z}$ of the observations is given by σ^2/n (An Introduction to Statistical Learning, 2021). The bagging algorithm is shown below:

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x).$$

Figure 2.1 - Bagging Algorithm

Where $f_{bag}(x)$ is the average of all the predictors, B is the number of bootstrapped training sets, $f^{*b}(x)$ is the average of the predictors in the b-th bootstrapped set. (An Introduction to Statistical Learning, 2021)

Random Forests uses bagging to build its decision trees for its model but when selecting its predictors, chooses a random sample of m predictors from the full set of p predictors. In doing this we can think of this process as decorrelating the trees, thereby making the average of the resulting trees less variable and hence more reliable (An Introduction to Statistical Learning, 2021).

2.2.2 Linear Support Vector Classifier

This model is a version of Support Vector Machines. Support Vector Machines or SVM's are a supervised learning model where data features are represented as points in space and mapped so the class labels are divided by a hyperplane. In mathematics, a hyperplane H is a linear subspace of a vector space V such that the basis of H has cardinality one less than the cardinality of the basis for V (DeepAI, 2019). For example, in two dimensions a hyperplane will be a line, in three dimensions it will be a plane, in general in n-dimensional space a hyperplane is n-1 space. A general case linear support vector classifier will solve this optimization problem:

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,\, M}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \quad \sum_{i=1}^{n} \epsilon_i \leq C,$$

Figure 2.2 - Linear Support Vector Classifier Maximization Problem

Where M is the width of the margin, $\epsilon_1,\ldots, \epsilon_n$ are slack variables, $\beta_0, \beta_1, \beta_2, \cdots, \beta_p$ are the coefficients of the hyperplane, $x_{i1}, x_{i2}, \cdots, x_{ip}$ are the observations, y_i are the labels, and C is a tuning parameter (An Introduction to Statistical Learning, 2021)

This optimization problem will create a hyperplane where observations with a hyperplane value referred to as f(x), is f(x) > 1 will be assigned the first-class label and hyperplane values f(x) < -1 will be assigned to the other class label. When the value of the hyperplane is equal to zero, f(x) = 0, this indicates that the observation is on the hyperplane. Support vector classifiers are a distinct type of support vector machine because when building their model they intentionally misclassify training points to allow for a more robust hyperplane during testing. Specifically, in the optimization problem, the slack variables account for misplacing the training labels.

2.2.3 Decision Trees

Another machine learning model we used was a Decision Tree Classifier. A Decision Tree Classifier is based on a decision tree regressor which outputs a quantitative label. There are two main steps to building a decision tree regressor, first you divide the predictor space — that is, the set of possible values for $X_1, X_2,\ldots,X_p$ — into J distinct and non-overlapping regions, R_1, $R_2,\ldots,R_J$. Then for every observation that falls into the region Rj , we make the same prediction, which is simply the mean of the response values for the training observations in Rj (An

Introduction to Statistical Learning, 2021). To determine the optimal regions, Rj, the residual sum of the squares is minimized. The residual sum of squares (RSS) is a method for measuring error within statistics, shown below it will take the square of the sum of each difference between the predicted value and the actual value (DataCadamia, 2020):

$$RSS = \sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

Figure 2.3 - Residual Sum of Squares Equation

Where y_i is the predicted value and $\hat{y}_i$ is the actual value (DataCadamia, 2020)

To find the split between the regions a greedy approach is used known as binary splitting, which goes step by step in the tree to create the splits. At each level the split is done at the value where a number of observations are strictly less than the split point, such that $\{X|Xj < s\}$ and a number of observations are greater than or equal to the split value such that $\{X|Xj \geq s\}$. (An Introduction to Statistical Learning, 2021). The split value is chosen by finding the one that minimizes RSS. This process of splitting is repeated for all j predictors. These are the general steps for building a decision tree regressor, however, in a decision tree classifier instead of RSS which is used for quantitative observations, a different error minimization is used. In classification, the output of the tree is the most often occurring label of the training observations. So therefore to measure the error in the step of dividing the regions, the fraction of the training observations in that region that do not belong to the most common class is used (An Introduction to Statistical Learning, 2021).

2.2.4 K Nearest Neighbors

K nearest neighbors is an instance-based learning or non-generalizing learning model (scikit-learn developers, 2021). This means that it does not create a model similar to other types of machine learning, it will simply remember the classification of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point (scikit-learn developers, 2021). Meaning given a positive integer K and a test observation x_0, the KNN classifier first identifies the K points in the training data that are closest to x_0,

represented by N_0. It then estimates the conditional probability for class j as the fraction of points in N_0 whose response values equal j: (An Introduction to Statistical Learning, 2021)

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

Figure 2.4 - Conditional Probability equation for class labels in KNN

Where Pr(Y = j | X= x_0) is the probability that given X = x_0 -> Y=j, K is the number of neighbors

(M.W Gardner and S.R Dorling, 1998)

The classification is then assigned to the label with the highest probability from the equation above.

2.2.5 Multilayer Perceptron

A multilayer perceptron (MLP) is a type of neural network which consists of a system of simple interconnected neurons, or nodes, which is a model representing a nonlinear mapping between an input vector and an output vector (M.W Gardner and S.R Dorling, 1998.). Between the input and output vectors are multiple layers of perceptrons, that all have input layers, activation functions, and output layers:

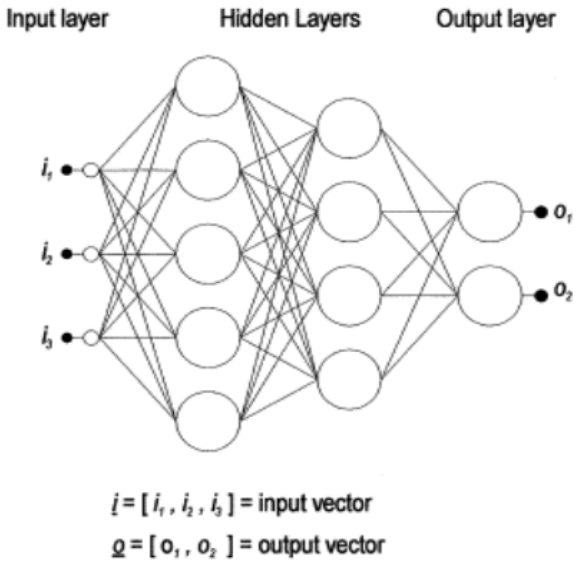

Figure 2.5 - Example Multi-Layer Perceptron

In this figure above, the input, hidden, and output layers of a MLP are shown (M.W Gardner and S.R Dorling, 1998)

The activation functions are a function of the sum of the inputs to the node modified by a simple nonlinear transfer (M.W Gardner and S.R Dorling, 1998.). Multilayer perceptrons work for classification problems by having the input layers be the data features and the output layers be the labels. The models are then formed through a binary labeling process, for example, if there were 3 labels then class label one would be represented as [1 0 0].

2.3 Current Framework

Past Fidelity interns have created code for executing and running the spend classification models. The models currently being used are Linear SVC and Random Forests. Linear SVC is a class of support vector machines (SVCs) - "a set of supervised learning methods used for classification, regression, and outliers detection" (scikit-learn, n.d.). Linear SVC performs multi-class classification on a dataset by using a "one-vs-the-rest" strategy. In a "one-vs-the-rest" strategy, the dataset is split into binary classification problems with each class being given a problem (Brownlee, 2020). A class membership probability score is given for each problem then the model assigns the class with the highest predicted probability. The other model, Random Forest Classifier, is a meta estimator and takes another estimator (decision tree classifiers) as a parameter (scikit-learn, n.d.). Decision trees can be thought of as a number of branches, where at each node a question is asked to sort a dataset. The results are subsets of data where the data points within each set are as similar as possible and the subsets are as different from each other as possible. In Random Forests, a decision tree is fitted to each sub-samples of the dataset and averaged to improve the predictive accuracy of the model. The implementation of multiple decision trees yields higher accuracy because the low correlation between the trees means a few errors in the model will not greatly affect the overall outcome of the prediction (Yiu, 2021).

2.4 Project Benefits

Correct spend classification is essential to maintaining Fidelity's business structure. Classified data entered into the procurement software can be used to help Fidelity make a multitude of key business decisions. For example, organized data can help analysts identify areas where Fidelity might reduce costs. If the cost reduction coincides with the need for more efficient processes, this analysis might also lead to process improvement. Having a leaner, or a more efficient, process will also help Fidelity reduce the cost of spend analytics. If the spend classification models take too long to produce results or produce inaccurate results, Fidelity will lose investments in time and money. Miscategorized spending could distort end-of-year profit, give an inaccurate representation of predicted spending for the next year, or create problems through improper documentation of spending (APQC, n.d.).

Fidelity also uses the procurement software to assist with finding vendors. Within the software, users can find stored data such as a vendor catalog, past prices, and external vendors. The vendor catalog allows users to find vendors used in the past, some of which might provide discounts. Using the catalog allows users to save time - rather than researching vendors and asking for quotes, they have access to approved vendors and past prices. Using these prices, the user can estimate what their cost might be and compare it to other vendors. The procurement software also provides a tool for searching new vendors for diversification and better savings (Weaver, 2021). The data going into this software must be accurate so that users can make intelligent decisions concerning vendor choice. Users should seek out quality products at the lowest price to reduce Fidelity's costs. Once the vendor is chosen, the decision will need to be justified and accepted before the purchase is made. Therefore, the user needs accurate data from the procurement software.

3. Methodology

3.1 What is Agile Scrum?

Software engineering teams need organization in their development process to allow for productivity to thrive and deliver the product they are developing in a timely manner. To accomplish these goals within our team, we used the Agile Scrum project management methodology. Agile is a methodology where continuous iterations and testing take place during the entire life cycle of a product (Apoorva Srivastava, Sukriti Bhardwaj, and Shipra Saraswat. 2017). There are three main roles of an Agile Scrum team: the scrum master, the product manager, and the developers. The scrum master maintains order and eliminates impediments across the development team. The product manager is the connection between the stakeholder and the rest of the team, communicating what the stakeholder wants from the deliverable.

3.1.1 Workflow of Agile Scrum

Agile scrum works over periods of time called sprints, which can last from one to four weeks. Over the course of our project, we had seven one week-long sprints. During the sprint, team members take tasks out of the product backlog and work individually or in smaller teams to complete them. The product backlog is a documentation of all the requirements to be worked on in the current sprint (Apoorva Srivastava, Sukriti Bhardwaj, and Shipra Saraswat. 2017). The information in the product backlog is discussed between the product manager and the stakeholder to create user stories. A user story is an informal, general explanation of a software feature written from the perspective of the end user. (Rehkopf Max) Team members are assigned user stories and from those develop code centered around delivering the product. At the end of each sprint, the objective is to produce a potentially shippable product. In figure 1, a diagram of basic scrum workflow is shown.

Figure 3.1 - Agile Scrum Workflow (Rehkopf Max)

3.1.2 Daily Scrum

Every working day on an Agile Scrum team, the team will meet along with the scrum master to have a discussion on the progress of work being done. Daily scrums keep members accountable and provide the group with a good idea of feature progression. The focus of internal scrum meetings is to answer 3 questions:

1. Since the last scrum, what have you worked on?
2. What are you going to work on today?
3. Do you foresee any problems with your work/have any concerns?

These questions aim to provide team members with individual updates on stories being worked on.

3.2 Sprints

Agile Scrum sprints are the set periods of time, typically one to four weeks in length, where predetermined tasks are worked on by dividing up the work amongst the team. Scrum revolves around sprints and is the central part of the methodology.

3.2.1 Sprint Planning

In order to work solely on user stories and tasks during the sprint, a sprint planning session is held before each sprint to organize the work being done. Planning is done with the whole team, it is a time to create user stories, assign tasks, and discuss expected outcomes of the sprint.

3.2.2 Sprint Review

Another important part of scrum is holding a sprint review. A sprint review is when the scrum team updates the product manager and stakeholders with progress made. This is primarily done through demonstrating the completed user stories that have been worked on during the sprint. These meetings serve as a place for the product manager and stakeholders to give feedback or comment on the work done.

3.2.3 Sprint Retrospectives

Reflecting on the success of sprints is the purpose behind holding retrospective meetings. These serve as a place for all team members to reflect on the good and bad of the previous sprint so that workflow and team cooperation are improved. We centered the meeting around everyone answering three questions: What did we do well? What can we do better? What will we commit to improving in the next sprint? The final question focuses on what we will commit to improving during the next sprint, after each retrospective we also kept track of which commitments we ended up reaching. If there were any we did not complete in the previous sprint that we still felt needed to be complete, we would add those to the new list we were compiling in that retrospective meeting.

3.3 Scikit-learn

Throughout the term, we implemented and tested multiple different machine learning models. To do this easily and follow directions from Fidelity, we used the Python machine

learning library scikit-learn. Scikit-learn or sklearn is widely used in the machine learning community for its simplicity, it allowed us to create all of our models and provided functions for testing accuracy and tuning hyperparameters. Hyperparameters are parts of the models that can be tuned, an example of this is with Random Forests where a hyperparameter is the number of trees in the model, and trees is the number of decision tree models created to make up the whole forest. To begin all of our models we would simply import the function needed for the model. To train all of our models we would call the fit function that takes in the training features and training labels. Finally, to score our models we would use a set of functions common for classification problems to give us a full report of the accuracy. These included: accuracy_score, precision_score, recall_score, classification_report, and confusion_matrix. The most important metric of these is accuracy_score as this represents the set of labels predicted for a sample that must exactly match the corresponding set of labels (scikit-learn). The other functions help us identify specific incorrect labeling which, for the purpose of our project, was not necessary to reach our goal. However, we left the other functions in our code since they caused no hindrance to our models. They were used by the previous team who worked on this project, and it leaves the option there if future teams choose to examine specific mislabelings.

4. Software Development Environment

4.1 Project Management Software

The development software used for our group's projects was determined by Fidelity Investments, including the version of the software.

4.1.1 Discord & Microsoft Teams

For team communication, we used Microsoft Teams on Fidelity's virtual machine and Discord. This software provides a convenient way for us to send messages amongst the team. Both Teams and Discord give the option to send messages to everyone on the team or communicate directly with another team member. They also have mobile and desktop apps for convenience. We used Microsoft Teams on Fidelity's virtual machine to communicate with Fidelity's colleagues and to discuss any company-related materials with each other. When communicating some daily arrangements and other team chores, we used Discord.

4.1.2 JIRA (Fidelity Company Portal & Team Portal)

JIRA is a project and transaction tracking tool from Atlassian. It is widely used in defect tracking, customer service, requirement gathering, process approval, task tracking, project tracking, and agile management (Atlassian, n.d.). Our team joined the Fidelity Jira group and performed our weekly sprint on the Fidelity Jira board. We also had a separate team Jira board for tasks that are not related to the company's process, such as paper editing, to keep track of our progress.

4.2 Programming Environment

4.2.1 Visual Studio Code 1.57.1

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code

completion, snippets, code refactoring, and embedded Git. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for programming languages (Microsoft, n.d.). We used Visual Studio Code to create and edit all Python scripts for our project.

4.2.2 Python 3.6.4 (Anaconda3 5.1.0)

Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS (Jetbrains, n.d.). In the preliminary section of our project, Anaconda was used to perform local tests of our own Python scripts for the purpose of familiarization of the code structure.

4.3 Software Tools

4.3.1 BMC Control-M 9.0.20.1000

Control-M simplifies application and data workflow orchestration on premises or as a service. It makes it easy to build, define, schedule, manage, and monitor production workflows, ensuring visibility, reliability, and improving service level agreements (a plain-language agreement between the provider and the customer that defines the services the provider will deliver, the responsiveness that can be expected, and performance measurements) (BMC Software, n.d.). The execution of all our Python scripts was heavily dependent on Control-M. It provides all necessary output including runtime, print statements, and log files for us to compare and analyze our machine learning models.

4.3.2 WinSCP 5.9.6

WinSCP is a free and open-source SSH File Transfer Protocol, File Transfer Protocol, WebDAV, Amazon S3, and secure copy protocol client for Microsoft Windows. Its main function is secure file transfer between a local computer and a remote server (WinSCP, n.d.). Our team

used WinSCP to connect the repository of code of execution of Control-M, modify files for Control-M jobs, and examine the log files if necessary.

4.4 Data Sources

Data was processed and transferred from one zone to another in Snowflake's Data Cloud. It is powered by an advanced data platform provided as Software-as-a-Service (SaaS). Snowflake enables data storage, processing, and analytic solutions that are faster, easier to use, and far more flexible than traditional offerings. To the user, Snowflake provides all of the functionality of an enterprise analytic database, along with many additional special features and unique capabilities (Snowflake, n.d.). For our project, Snowflake was the primary source of data. All the spend data used for training and testing are obtained from the Snowflake table via our Python scripts.

5. Software Requirements

The project requirements were gathered through meetings with the sponsors at Fidelity. Since Fidelity is a large corporation, they already have strict guidelines for what software to use for functions such as scripting, Agile, development, etc. After the initial sponsor meeting, the software requirements were introduced but not fully elaborated. The second sponsor meeting included a presentation where the managers explored two different focuses that the team could pursue based on preference. The first focus was improving the performance of models with multithreading and enhancing the spend classification correction form. The second focus was analyzing spending and building a forecasting model.

5.1 Software Requirement Gathering Strategy

The team decided to meet after the second sponsor meeting to decide which of the two tasks to work on and exactly how the software requirements will play out. Of the two focuses the team was given, we went with the former focus due to its bigger impact on the investment firm as well as piquing the group's interests more. Discussions in the group meeting also included which pieces of software needed to be installed and what should be reviewed (for example, Python and Angular are big parts of the project, so we spent time familiarizing ourselves with both).

5.2 Functional and Non-Functional Requirements

Software requirements are typically divided into two different categories: functional and non-functional requirements. A functional requirement "describes the functions a software must perform". In other words, the concrete features of the software. Non-functional requirements, however, are defined as "the set of standards used to judge the specific operation of a system". Non-functional requirements deal with aspects such as performance, usability, readability, and other quality of life changes that are not necessarily features (Martin, 2021).

From the second meeting the team had with the sponsors, the below requirements were given in a presentation:

- To improve performance of the models using Python multithread on Amazon EMR
- To enhance the Spend Classification Correction Form using Angular, NodeJS, REST API, and Python Flask

The first focus dealt with optimizing the classification algorithm by using different models, optimizing models, and leveraging Python multithreading. We researched and implemented the following models: KNN, Gaussian process, and decision trees. Adding these new models, as they are essentially new features, would be classified as a functional requirement. The multithreading was attempted by delegating different threads to different tasks of the pipeline, such as parts of the data fetching, pre-processing, training, and testing being done asynchronously. Thus, it falls under the category of a nonfunctional software requirement. More specifically, the task asked for reducing the time it takes to run the pre-existing machine learning models (random forests and Linear SVC) and any models we added. The data that was used was provided by the investment firm, so a remote desktop was used to benchmark the models.

The second focus was on adding features to a pre-existing classification correction form that is built with technologies such as Angular and NodeJS. We did not complete this task due to time constraints, but we did plan out the requirements. The current form only allows users to edit the level three commodity classification, but would be more useful with additional editing selections. The level 3 commodity was defined by a set of selection criteria. If one criterion was wrong, the result was a data point that had been classified to the wrong level 3 commodity. In this case, simply editing the commodity did not resolve the issue. The ability to edit selection criteria would have needed to be added to the correction form in order to remedy this. As this requirement was dealing with programming concrete features to the angular application, it would have been categorized as a functional requirement.

5.3 Epics and User Stories

The team followed Agile Software Development to manage our project. Agile employs user stories to organize tasks the team is to complete. User stories are defined as “an informal, general

explanation of a software feature written from the perspective of the end user or customer" (Rehkopf, n.d.). They describe how the exact functional/non-functional requirement will provide value for a certain user, whether that be a customer or an employee. User stories are very useful in that they provide a guideline on what is needed per feature, so it is easier to plan how much progress can be made per sprint. They are also very flexible and can easily change throughout agile development.

An epic is defined as "a series of user stories that share a broader strategic objective" (ProductPlan, 2020). In other words, an epic is a collection of user stories that relate to each other and, when combined, accomplish a larger task. Epics play an important role in keeping the tasks organized and manageable.

Below are the stories that we used throughout the project:

Epic 1: Initial Research and Preparations

a. As a Fidelity employee
    i. I want to make sure that those working under me are set up for success so that problems can be avoided
b. As a student
    i. I want to have a solid understanding of the project so that I can produce meaningful deliverables
c. As a developer
    i. I want to download necessary software and read the scripts so that I can be prepared for the following weeks

Epic 2: Enhance Spend Classification non-NLP AI/ML Framework

a. As a developer
    i. I want to research and implement decision trees so that we can compare it to other models
    ii. I want to identify the most important aspects of a data set so that I can

eliminate the least important ones

iii. I want to run the non-nlp multi thread models on the largest dataset

iv. I want to multithread KNN so that the run time is reduced

v. I want to research and implement KNN (K Nearest Neighbors) so that we can compare it to other models

vi. I want to improve the pre processing methods used so that run time is decreased and accuracy is increased

vii. I want to be able to efficiently use Control-M so that I can collaborate with my teammates to run Python scripts

viii. I want to research and implement gaussian process so that we can compare it to the other models

ix. I want to optimize the hyperparameters of the machine learning model so that we get higher accuracy

b. As an analyst

i. I want to be able to visualize spend data post processing so that I can identify trends

Epic 3: Enhance Spend Classification NLP AI/ML Framework

a. As a developer

i. I want to run the NLP multi thread models on the largest dataset

ii. I want to fine tune the preexisting NLP so that our models are more accurate

iii. I want to make decision trees and random forests use NLP so that we can measure their accuracy

iv. I want to multithread decision trees so that the run time can be reduced

v. I want to document my project so that future work can be easily continued

vi. I want to fine tune a MLP so that we can have a high accuracy model

vii. I want to run all our models on the largest data set so that I can see the run time and accuracies

viii. I want to create a document containing the accuracies of all our models
ix. I want to clean up the FCT_SPEND_add_multithread_param file

b. As an analyst
i. I want to research NLP models used previously so that we can analyze if they are more accurate than non-NLP

Epic 4: Project Paper

c. As a student
i. I want to complete editing for the software requirements section of the paper.
ii. I want to finish the writing for the design section of the paper.
iii. I want to add an assessment section to the paper.
iv. I want to add an acknowledgements section of the paper.
v. I want to add a future work section to the paper.
vi. I want to edit every section of the paper.
vii. I want to use a citation tool to complete in line citations for the paper.
viii. I want to add/edit in a table of contents.
ix. I want to add machine learning backgrounds for the various techniques we used.
x. I want to add a machine learning methodology section for the library sklearn.
xi. I want to turn our workflow chart into a UML activity diagram.
xii. I want to add a technical learning section.
xiii. I want to add a burndown chart to the software design section

6. Software Design

6.1 System Architecture

To properly construct Python script files and implement the machine learning models for spend classification, we had to understand the architecture of the company's spend classification process and where our files are executed in terms of the whole system. As seen in Figure 6.1, we learned that the spend data is loaded from the Oracle feed files to Snowflake Database tables for processing. Then the process is divided into four zones in sequence to handle the data via Control-M until the data processing is completed. As a team, our job was to determine the most accurate and efficient machine learning model for spend classification. The first zone (Raw Zone), which is circled in red in the figure, was our main focus since all the machine learning models are trained and tested in this zone. By fetching the spend data from the Snowflake tables, we were able to perform a preliminary pre-processing of the data and apply the supervised learning technique with different spend machine learning models. Such a technique uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time (IBM, n.d.). The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized, and is suitable for the next process.

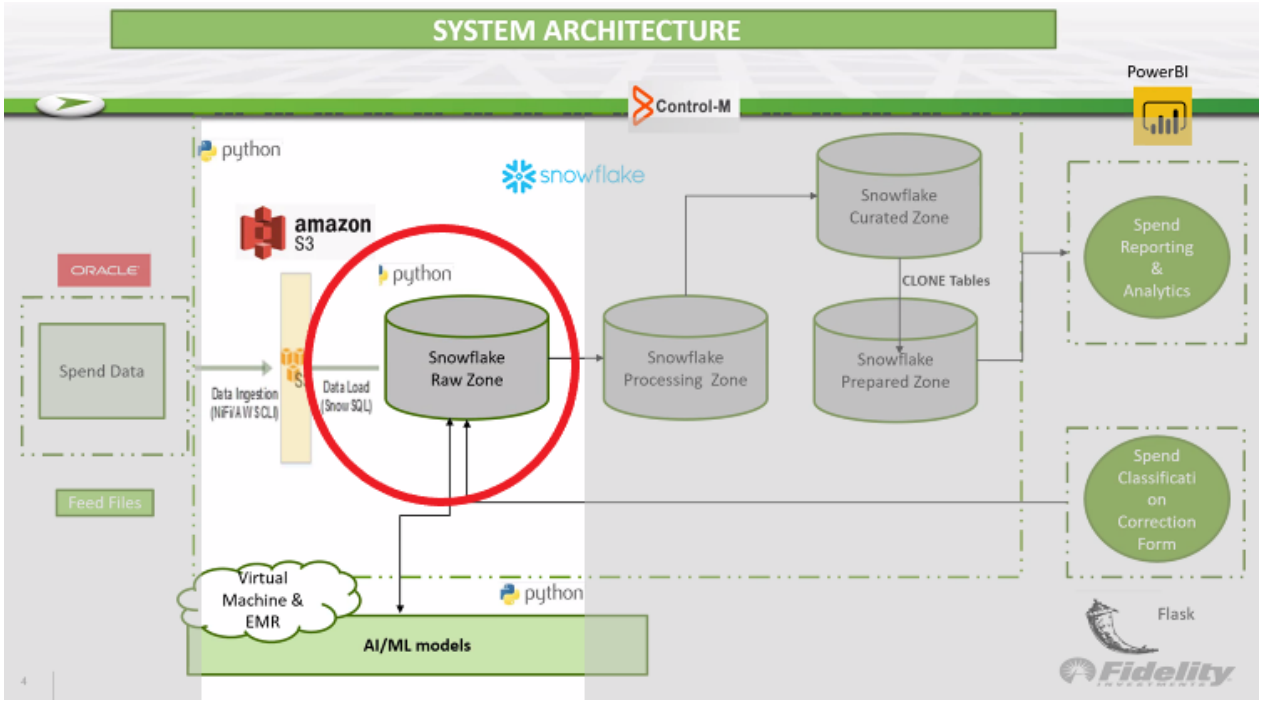

Figure 6.1 - System Architecture Diagram (Corporate Technology Group, 2021)

6.2 Snowflake Database Usage

The first section of our code in the Python script files is data fetching and pre-processing. As mentioned previously, spend data is loaded into Snowflake tables. The two tables we used for development have the same schema, the difference lies in the amount of data they store: the first one only stores 10,000 rows of spend data for more of a testing purpose; the second table is used to simulate the actual operation so it stores 5,000,000 rows of data, which is closer to the approximate 9,000,000 rows in the production environment. Although there are 92 table entities for each table, only a few (less than 10) entities are selected in data pre-processing, such as FACT_ID, COST_CENTER_DESC, GL_ACCOUNT_DEC, VENDOR_NAME, etc. These are the entities that have the most impact on spend classification both on the business side and the programming side. The final result of the classification is the generation of SUB_COMMODITY_ID for each spend datapoint.

6.3 High-Level Development Workflow

At the beginning of our project, we studied the code structure of the Python machine learning scripts written by the previous procurement team. This is shown in Figure 6.2 below, which demonstrates our workflow from a high-level perspective. After conducting a thorough study of machine learning models, we implemented three new machine learning models and performed tests and analysis on these models with small and large size datasets. We also implemented both natural language processing and non-natural language processing techniques to the pre-processing section of our code to compare the results along the way. By examining the result, we were able to determine the best machine learning models in terms of accuracy and run-time efficiency, and multi-threaded these models to check if the runtime decreased. At the end of our project, the deliverable consisted of the fully developed models, and corresponding reports were formed and submitted to the company.

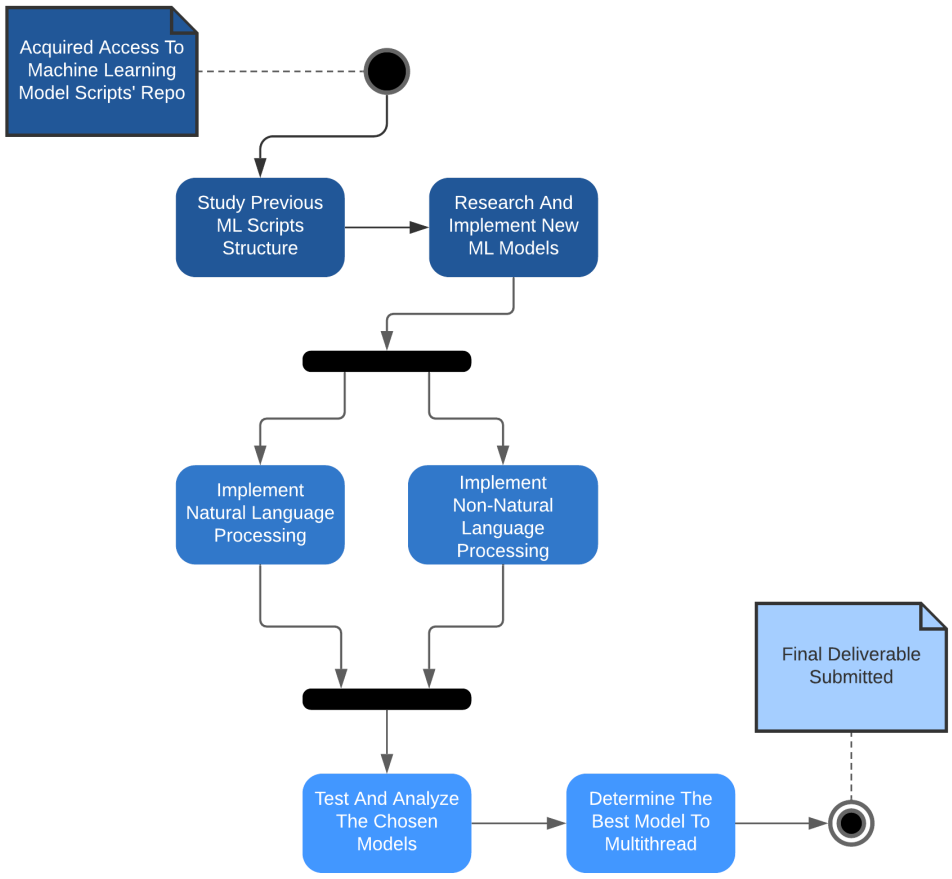

Figure 6.2 - High-Level UML Activity Diagram of Development Workflow

6.4 Machine Learning Model Investigation Process

To ensure the accuracy and standardization of the training and testing of our models, we strictly followed the company's development and testing process with the specific software requirements.

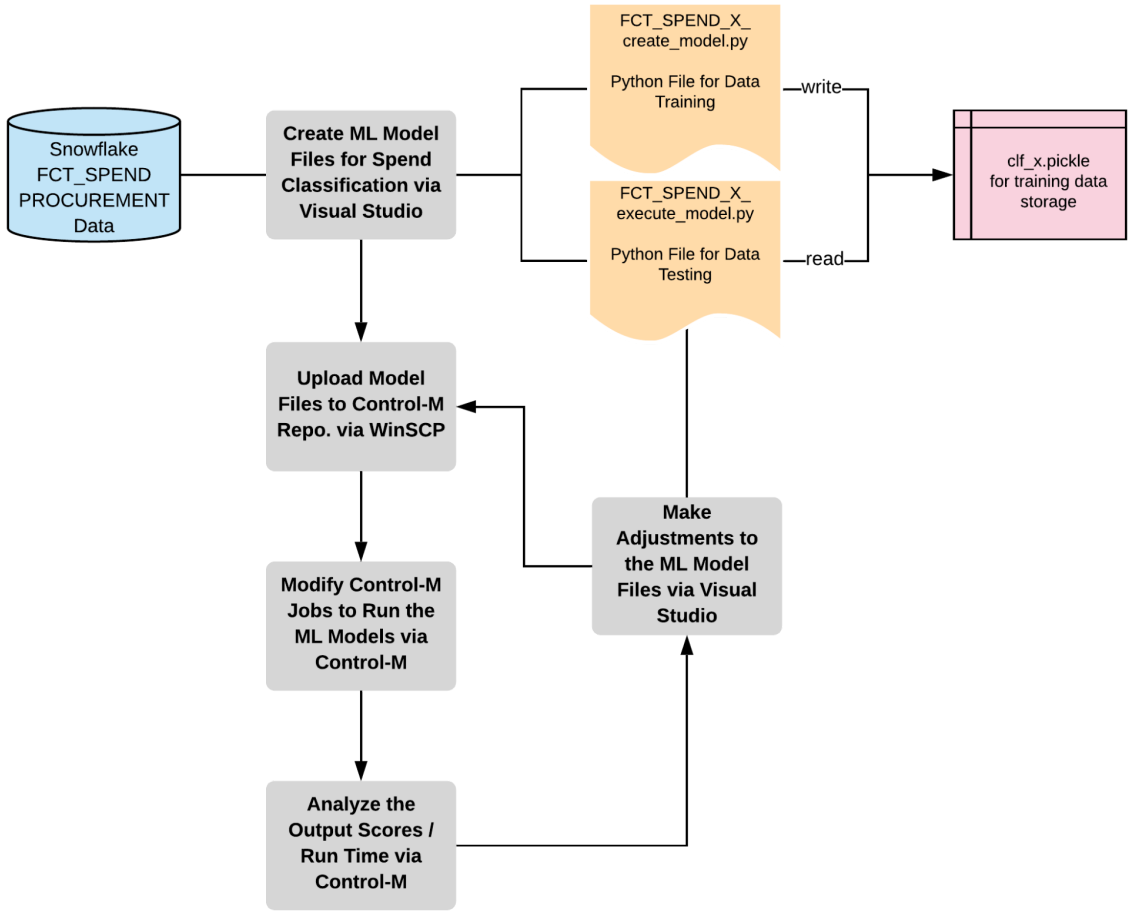

Figure 6.3 - ML Model Investigation Process Diagram

We first created the machine learning model file in Python with our local IDE (Visual Studio), as shown in Figure 6.3. These models included but were not limited to K Nearest Neighbor, Decision Tree, Gaussian Process, etc. For each machine learning model, we created two files: create_model.py for data training and execute_model.py for data testing. Both files contained the same process of fetching spend procurement data from the company's Snowflake

Database, and the same data pre-processing method, which is with or without the natural language processing. The create_model.py trained the model after data pre-processing, displayed the training score and time consumption, and stored the model in a .pickle file with the name of the model. The execute_model.py then read the trained model from the .pickle file and displayed the result including the accuracy score after testing the selected model with the spend procurement data. When both files were created, we uploaded them to the vl4117rtp2 repository of Control-M via WinSCP. Then, in Control-M, we ordered the corresponding job to run the create_model.py and execute_model.py in order and examined the output of the execution. We optimized and improved the existing model code in our IDE based on the analysis of the execution output, and then re-followed the previous steps to upload and test the model.

7. Software Development

Throughout the term our team held seven sprints, each focusing on different outcomes given our situation in our development process. During each sprint, we met as a team Monday through Friday with two employees from Fidelity (our manager and squad leader) as optional participants. Each of our meetings centered around the progress we had made since our last meeting and functioned as our daily scrum meetings. Our manager, Michael, came to our meetings on a daily basis to stay up to date on any day-to-day progress we made. During the term, our team held our sprint planning meetings on Mondays in the morning, typically lasting two hours. Our sprint retrospective meetings were held at the end of every week on Friday, at the end of the workday, typically at 4 pm. Instead of having a set meeting for a sprint review, Fidelity found it best to have our manager with us in each daily scrum, so the progress reports were made then. After each team member finished their part of scrum, we would update Michael or let him know about any roadblocks we were facing. This was the general process that we followed throughout B term, however, we did have meetings sometimes during the day and the developers on the team met in person a few times. Keeping a structure and following the agile scrum process was part of what led to the success of our project.

7.1 Sprint 1: 10/25 - 10/30

In our first sprint, the team focused on setting up our virtual desktops and familiarizing ourselves with the project tasks. To complete our first story, the team needed to finish required onboarding training and attend meetings intended to help us learn how to set up our virtual machines and necessary software for the project. To complete our second story, we all researched Agile methodology so we could correctly implement it into our project. The computer science majors on the team also researched necessary software and subject matter that was important to the creation of our deliverables. To complete our final story, team members needed to download software and read Fidelity provided code and resources. The team was able to complete all of these tasks during the first week.

User Story	Subtasks	Points	Assignee(s)	Completion Status
As a Fidelity employee, I want to make sure that those working under me are set up for success so that problems can be avoided	-Attend Fidelity meetings setup by Michael -Complete training modules	2	All	Done
As a student, I want to have a solid understanding of the project so that I can produce meaningful deliverables	-Read documentation from past interns -Research current models used -Research Agile	2	All	Done
As a developer, I want to download necessary software and read the scripts so that I can be prepared for the following weeks	-Download software -Read scripts from past interns	2	Neville Luke Ziqian	Done

Table 7.1.1 - User Stories in Sprint 1

Total Story Points	6

Table 7.1.2 - Total Story Points in Sprint 1

We were able to take some important learnings from this sprint. First, we learned the difficulty of communicating within a large company. To reduce confusion on our end, we were provided with a manager to keep track of the project for Fidelity and provide us guidance. Our lead was new to the project but took strong initiative to help answer our questions. We quickly found that our previous understanding of the project was misled. Neville noted that the code was already multithreaded, a task which we thought we were meant to take on. Our manager helped set up a meeting so we received help promptly and were able to move on.

The team also quickly learned the benefits of Agile methodology. Each member had a different level of comfort and understanding going into the project. From our research and first sprint planning meeting, we were all able to get caught up to speed and understand the flow of Agile. We discovered the first benefit to Agile within our scrum meetings. The team found scrum meetings a good opportunity to catch up with each other. Within our scrum meetings notes template, we kept a notes section for each meeting. These notes contained the announcements and questions so that we could review them later and make sure they were all addressed. This made our retrospective meeting easier because we had extensive documentation of our week. The second benefit came with the implementation of user stories and epics. These made it very easy to track what needed to be accomplished in order to produce our deliverables and divide tasks between the team.

7.2 Sprint 2: 10/31 - 11/6

In our second sprint, the team focused on researching and implementing different machine learning models. In figure 7.2, the stories for this sprint are shown. Each of the three developers took the initiative on one of the three models in our user stories. On their chosen model, the developer first had to create the model files necessary to run the code through Control-M. Once completed, they compared the time to run and accuracy of the model against other models and stored the results. Accomplishing this took much longer than the team had hoped because of difficulties getting access to and running code through Control-M. Once we were able to finalize our results, our chosen models did not provide better results than Fidelity's current LinearSVC model (93% on EMR). Our lack of success required us to shift some of our focus towards NLP to see if we can get higher accuracy with our models moving into the following week.

User Story	Subtasks	Points	Assignee(s)	Completion Status

As a developer, I want to research and implement KNN (K Nearest Neighbors) so that we can compare it to the other models	-Create the model files needed to run the jobs in Control-M based on previous files -Create/identify the Snowflake tables needed to store the results	3	Neville	Done
As a developer, I want to research and implement decision trees so that we can compare it to the other models	-Create the model files needed to run the jobs in Control-M based on previous files -Create/identify the Snowflake tables needed to store the results	3	Luke	Done
As a developer, I want to research and implement gaussian process so that we can compare it to the other models	-Create the model files needed to run the jobs in Control-M based on previous files -Create/identify the Snowflake tables needed to store the results	3	Ziqian	Done
As an analyst, I want to be able to visualize spend data post processing so that I can	-Use PowerBI to visualize the spend data by connecting	3	Neville	Transfer to next sprint

identify trends	Snowflake to PBI desktop -Plot the important features in various different groups VS sub_commodity_id to help identify relationships -Analyze and compare the graphs to identify trends and help choose and appropriate model			
As a developer, I want to identify the most important aspects of a data set so that I can eliminate the least important ones	-Write code to verify the top features chosen -Identify why only six of the features were chosen to be analyzed via chi2 contingency	3	Luke	Transfer to next sprint
As a developer, I want to be able to efficiently use Control-M so that I can collaborate with my teammates to run Python scripts	-Work with Fidelity employees to learn how to use Control-M -Work with teammates to solves issues with running the code	5	Neville	Transfer to next sprint

Table 7.2.1 - User Stories in Sprint 2

Total Story Points	20

Table 7.2.2 - Total Story Points in Sprint 2

As a result of the lost time, the team was not able to finish the three other user stories we had set out to complete during this sprint. We were able to accomplish parts of the two of the stories, and will be able to finish them in sprint 3. After discussion, some confusion surrounding the subtasks of the third user story to be transferred to the next sprint was smoothed out and we expected to also finish this story in the next sprint.

The team focused a lot on learning within our team this week. We wanted to make improvements to work more efficiently now that we have lost some time. First, we learned we need to be more proactive in seeking out potential issues and bringing them up as soon as possible. The sooner problems are fixed, the sooner we can move on. Second, we should collaborate more with our user stories. A theme came up with stories taking longer to finish because one team member did not have all the knowledge necessary to complete the task. If we can teach each other rather than having to learn from online sources, we can solve problems faster. Although each user story can only be assigned to one individual, there is nothing stopping us from working together.

As the team began to work on their chosen models, we realized the risk that one of the models used could be not as fast or accurate as a model that was not chosen. Fidelity asked the programmers to choose one model to research and implement each, so we were limited to three models when there are many more to choose from.

7.3 Sprint 3: 11/7 - 11/13

In the third sprint, the team focused on finishing our work with the non-NLP models and started to work with NLP models. There were several user stories concerning the non-NLP models that were left unfinished because Fidelity employees were not able to provide the team assistance this week. For example, we were prepared to visualize the spend data but did not have access to the Snowflake tables for PowerBI yet so we were unable to finish the user story. There are plans to resolve issues with EMR and control-M on Monday, so we will be able to finish

stories involving this software early in the next sprint. To continue working while we waited, the team added several user stories at the end of the week that will continue into the next sprint.

User Story	Subtasks	Points	Assignee(s)	Completion Status
As an analyst, I want to be able to visualize spend data post processing so that I can identify trends	-Use PowerBI to visualize the spend data by connecting Snowflake to PBI desktop -Plot the important features in various different groups VS sub_commodity_id to help identify relationships -Analyze and compare the graphs to identify trends and help choose and appropriate model	3	Neville	Transfer to next sprint
As a developer, I want to identify the most important aspects of a data set so that I can eliminate the least important ones	-Write code to verify the top features chosen -Identify why only six of the features were chosen to be analyzed via chi2 contingency	3	Luke	Transfer to next sprint
As a developer, I want to be able to efficiently use	-Work with Fidelity employees to learn how	5	Neville	Done

Control-M so that I can collaborate with my teammates to run Python scripts	to use Control-M -Work with teammates to solves issues with running the code			
As a developer, I want to improve the preprocessing methods used so that run time is decreased and accuracy is increased	-Change the LabelEncoder to a OneHotEncoder -Remove print statements -Measure the time to run on certain steps	1	Neville	Transfer to next sprint
As a developer, I want to optimize the hyperparameters of the machine learning model so that we can get higher accuracy	-Write code to tune the hyperparameters of the machine learning model	3	Ziqian	Transfer to next sprint
As an analyst, I want to research NLP models used previously so that we can analyze if they are more accurate than non-NLP	-Read completed guide on NLP models -Look at all past NLP methods used by previous interns to understand NLP models and if they are more accurate than non-NLP	3	Ziqian	Transfer to next sprint
As a developer, I want to make	-Take the current NLP	1	Luke	Done

decision trees and random forests use NLP so that we can measure their accuracy	code and apply it to decision trees and random forests -Output the test accuracy			
As a developer, I want to fine tune a MLP so that we can have a high accuracy model	-Write code to tune the hyperparameters of MLP -Research neural networks	5	Neville	Transfer to next sprint
As a developer, I want to fine tune the preexisting NLP so that our models are more accurate	-Research the methods used by previous interns -Fine tune hyperparameters	3	Luke	Transfer to next sprint

Table 7.3.1 - User Stories in Sprint 3

Total Story Points	27

Table 7.3.2 - Total Story Points in Sprint 3

The stories the team added revolved around increasing the accuracy of our models using NLP and MLP. Our previous work with non-NLP models gave us unsatisfactory results, which led us to pursue different training algorithms. The team was mainly focused on NLP models, as past Fidelity interns have done with some success in terms of accuracy. To do this, we started with increasing our knowledge and understanding of what past interns have done so that we can improve on their previous work. This included optimizing the hyperparameters to increase accuracy and measuring the accuracy of decision trees and random forests using NLP. In addition, the team will be implementing a MLP to see how the accuracy of this model compares to current options.

As we finished our first full week working with Jira, we were thinking about how we could use the product more efficiently. We noticed even though we were keeping track of our stories in Jira, we were still communicating our accomplishments like we were using our previous scrum setup. This did not transfer well and led to some miscommunication on user stories and subtasks. Moving into the next sprint, we need to make scrum revolve around user stories as presented in the Jira and all team members need to individually check the Jira everyday. To further improve the team's efficiency we considered what other process improvements we can make. We agreed that we would like to be able to resolve technical issues faster. To do this we will need to clarify our project timeline to Fidelity project managers and eliminate issues caused by the difference in our sprint schedule.

In our third sprint, the team started to see a pattern of slow response time. Although our project manager is doing a good job acknowledging our problems, responses from those being asked to help are slow. This difficulty might be caused by the difference between our sprint pace and Fidelity's sprint pace. For us, it is important that problems are solved immediately so that we can accomplish all our tasks in one week. Having two week sprints means that Fidelity can easily push unaccomplished tasks to the next week. Many employees have also been absent to celebrate Diwali, making it hard to schedule meetings. We expect this will continue to occur as more holidays are ahead and employees will be trying to use any vacation time that does not transfer into the new year.

7.4 Sprint 4: 11/14 - 11/20

Sprint 4 continued to move slowly for the team, but some important discoveries were made. The main contributor to the team's slow pace was a lack of access to PowerBI and EMR. Through investigation with our manager, we learned and completed the process for requesting access to the software. Unfortunately gaining access could take up to two weeks, so the team may not be able to use PowerBI or EMR before the end of the term. Without PowerBI and EMR access, the team focused on improving the current code which led to an important discovery. The

testing and training data were not split, resulting in the same accuracies. Once the data was split the team was able to get the correct results for accuracy.

User Story	Subtasks	Points	Assignee(s)	Completion Status
As an analyst, I want to be able to visualize spend data post processing so that I can identify trends	-Use PowerBI to visualize the spend data by connecting Snowflake to PBI desktop -Plot the important features in various different groups VS sub_commodity_id to help identify relationships -Analyze and compare the graphs to identify trends and help choose and appropriate model	3	Neville	Transfer to next sprint
As a developer, I want to identify the most important aspects of a data set so that I can eliminate the least important ones	-Write code to verify the top features chosen -Identify why only six of the features were chosen to be analyzed via chi2 contingency	3	Luke	Transfer to next sprint
As a developer, I want to improve the preprocessing	-Change the LabelEncoder to a	1	Neville	Transfer to next sprint

methods used so that run time is decreased and accuracy is increased	OneHotEncoder -Remove print statements -Measure the time to run on certain steps			
As a developer, I want to optimize the hyperparameters of the machine learning model so that we can get higher accuracy	-Write code to tune the hyperparameters of the machine learning model	3	Ziqian	Done
As an analyst, I want to research NLP models used previously so that we can analyze if they are more accurate than non-NLP	-Read completed guide on NLP models -Look at all past NLP methods used by previous interns to understand NLP models and if they are more accurate than non-NLP	3	Ziqian	Done
As a developer, I want to fine tune a MLP so that we can have a high accuracy model	-Write code to tune the hyperparameters of MLP -Research neural networks	5	Neville	Transfer to next sprint
As a developer, I want to fine tune the preexisting NLP so	-Research the methods used by previous	3	Luke	Transfer to next sprint

that our models are more accurate	interns -Fine tune hyperparameters			

Table 7.4.1 - User Stories in Sprint 4

Total Story Points	21

Table 7.4.2 - Total Story Points in Sprint 4

The team did not add any new stories this week to focus on resolving stories from the past week. We ended up making many improvements as a team. Frustrated with remote work, the team members living on campus met in person to work. They found this to greatly increase communication and productivity. As a result, we are planning to meet in person a couple times a week for the rest of the project. The team also started finding a better flow for our scrum meetings, which we will continue to improve upon in the following week. We found that centering scrum around our user stories helped maintain focus and communicate progress. This will continue to be done as well as saving discussion for the end of scrum. Pausing to have discussions while going through the user story progress made it hard for the scrum master to track important information.

A risk that has become more clear as the project progresses is that we may not be able to run our code on EMR. If this happens, Fidelity might not accept our findings so readily. Next week the team hopes to move forward with running the code on our machines with a larger dataset without EMR. We had not done this yet because we were not sure if a larger data set could be supported. Our manager seems confident that we should pursue this despite the uncertainty. We also hope to mitigate this risk by seeing if a Fidelity employee could run our code using their EMR access.

7.5 Sprint 5: 11/21 - 11/27

Because of the Thanksgiving holiday, the team decided not to pursue any user stories focused on our code this week. Instead, we focused on improving our paper. With the deadline

for the final draft quickly approaching, we found it important that we take the time to get the proposal as close to being done as possible.

User Story	Subtasks	Points	Assignee(s)	Completion Status
As a student, I want to complete editing for the software requirements section of the paper.	-Complete writing and editing for the software development section	1	Luke	Done
As a student, I want to finish the writing for the design section of the paper.	-Complete writing and framework for the design section	3	Ziqian	Done
As a student, I want to add an assessment section to the paper.	-Complete writing for the assessment section	3	Olivia	Done
As a student, I want to add an acknowledgements section of the paper.	-Add a complete acknowledgements section	1	Neville	Done
As a student, I want to add a future work section to the paper.	-Complete writing for the future work section	2	Neville	Done
As a student, I want to edit every section of the paper.	-Every team member adds edits to the paper	3	All	Continue to the end of sprint 7
As a student, I want to use a citation tool to complete in line	-Each team member adds in line citations to	1	All	Transfer to next sprint

citations for the paper.	their writing			
As a developer, I want to rerun my models on the new table.	-Run KNN, MLP, Decision trees, random forests, Linear SVC -Where applicable run with and without NLP	1	Luke, Neville, Ziqian	Done

Table 7.5.1 - User Stories in Sprint 5

Total Story Points	15

Table 7.5.2 - Total Story Points in Sprint 5

Good progress was made on the paper, which will be continued in sprint 6. The team also spent some time rerunning our models using a larger dataset with 5 million rows. From our results, we found that the accuracies of our models were not significantly greater than the pre-existing models'. Currently KNN without NLP and decision trees with NLP are the two models we feel most confident moving forward with multithreading.

As a team, we chose not to hold a retrospective meeting this sprint. Only formally working two days gave us very little material to work with. Instead the retrospective meeting at the end of sprint 6 will include work done in sprint 5.

7.6 Sprint 6: 11/28 - 12/4

In sprint 6, our focus revolved around beginning to finalize our project. We began the week by canceling user stories that were no longer beneficial to our progress. These two user stories were intended to help the team better understand the provided data set so that we could choose appropriate models and fine-tune them to produce the best results. Now that we have decided to pursue KNN without NLP and decision trees with NLP, we do not believe that these user stories are necessary.

User Story	Subtasks	Points	Assignee(s)	Completion Status
As an analyst, I want to be able to visualize spend data post processing so that I can identify trends	-Use PowerBI to visualize the spend data by connecting Snowflake to PBI desktop -Plot the important features in various different groups VS sub_commodity_id to help identify relationships -Analyze and compare the graphs to identify trends and help choose and appropriate model	3	Neville	Canceled
As a developer, I want to identify the most important aspects of a data set so that I can eliminate the least important ones	-Write code to verify the top features chosen -Identify why only six of the features were chosen to be analyzed via chi2 contingency	3	Luke	Canceled
As a developer, I want to improve the preprocessing methods used so that run time is decreased and accuracy is	-Change the LabelEncoder to a OneHotEncoder -Remove print	1	Neville	Done

increased	statements -Measure the time to run on certain steps			
As a developer, I want to fine tune a MLP so that we can have a high accuracy model	-Write code to tune the hyperparameters of MLP -Research neural networks	5	Neville	Done
As a developer, I want to fine tune the preexisting NLP so that our models are more accurate	-Research the methods used by previous interns -Fine tune hyperparameters	3	Luke	Transfer to next sprint
As a developer, I want to multithread decision trees so that the run time can be reduced	-Multithread the data fetching and the model	2	Luke	Transfer to next sprint
As a developer, I want to document my project so that future work can be easily continued	-Update guide based on our learnings/tasks	4	Neville	Transfer to next sprint
As a developer, I want to multithread KNN so that the run time is reduced	-Multithread the fetching of data and the model	2	Neville	Transfer to next sprint
As a developer, I want to clean	-Go through the file	1	Ziqian	Done

up the FCT_SPEND_add_multithread _param file	and remove print statements			
As a developer, I want to create a document containing the accuracies of all our models	-Create a document containing: the accuracies of all models, the run time of all models	1	Ziqian	Done
As a student, I want to edit every section of the paper.	-Every team member adds edits to the paper	3	All	Continue to the end of sprint 7
As a student, I want to use a citation tool to complete in line citations for the paper.	-Each team member adds in line citations to their writing	1	All	Done
As a student, I want to add a conclusion section to the paper.	-Write the conclusions for the paper	2	Luke	Done
As a student, I want to add/edit in a table of contents.	-Add a table of tables and table of figures	1	Ziqian	Complete at the end of sprint 7
As a student, I want to add machine learning backgrounds for the various techniques we used.	-Write background for all techniques used	2	Neville	Transfer to next sprint
As a student, I want to add a machine learning methodology	-Add a section on sklearn to the	1	Neville	Transfer to next sprint

section for the library sklearn.	methodology section			
As a student, I want to turn our workflow chart into a UML activity diagram.	-Change format of workflow chart to UML activity diagram	1	Ziqian	Done
As a student, I want to add a technical learnings section.	-Add technical learnings to the assessment section	2	Luke	Transfer to next sprint
As a student I want to add a burndown chart to the software design section	-Create a sprint-by-sprint burndown chart	1	Olivia	Done

Table 7.6.1 - User Stories in Sprint 6

Total Story Points	39

Table 7.6.2 - Total Story Points in Sprint 6

As decided in sprint 5, we created user stories for multithreading KNN and decision trees. Multithreading was not finished in sprint 6 because we found that when KNN and decision trees were multithreaded, their accuracies were reduced. This discovery was not made until the end of the sprint, so we have decided to transfer these stories to sprint 7 where we hope to resolve the issue. We also created user stories for documenting our work throughout the term for Fidelity's reference. We will be providing a guide outlining required infrastructure, our script files, flowcharts for NLP and non-NLP, how to run the scripts, and key concepts. This is intended to help with knowledge transfer if the project is continued. The team is also providing a table containing the accuracies and run times of all the models we worked on in this project for future reference. Lastly, we created user stories outlining some final additions to our paper.

In our retrospective, the team found that we are doing a good job staying focused and on pace. Our velocity in sprint 5 and 6 was greatly increased from past sprints. This puts us on track to complete our project in sprint 7. We have committed to continuing to improve our workflow in order to ensure that all user stories are complete by the end of sprint 7.

The team's most worrisome risk in this sprint was that we will not be able to multithread KNN and decision trees while receiving the best accuracy from the models. We made a plan to continue our user stories affected by this in sprint 7 in the hopes that we will be able to find a way to resolve the issue. If this is not possible, the current state of the multithreading will be added to our guide for Fidelity and in future work so that progress can continue to be made on the issue.

7.7 Sprint 7: 12/5 - 12/11

The final sprint of our project was eventful, as we needed to finalize our deliverables. The main focus of the week was editing and finalizing the paper to be sent for approval by Fidelity. Alongside this, the team also spent a lot of time rerunning code to record our final accuracies and run times. We also gathered materials for our final presentations to Fidelity. The first of these presentations will be a knowledge transfer to our manager, the division's squad leader, and two software engineers important to the project in which we will need to present our code and final results. The second presentation has a high-level focus to communicate what we had done throughout the project to our advisors and those who are interested at Fidelity. For this presentation, we created a user story to produce a PowerPoint.

User Story	Subtasks	Points	Assignee(s)	Completion Status
As a developer, I want to fine tune the preexisting NLP so that our models are more accurate	-Research the methods used by previous interns -Fine tune hyperparameters	3	Luke	Done
As a developer, I want to multithread decision trees so that the run time can be	-Multithread the data fetching and the model	2	Luke	Done

reduced				
As a developer, I want to run the NLP multithreaded models on the largest dataset	-Complete runs on all models: KNN, LSVC, RF, MLP, DT and give scores and time to run	2	Luke	Canceled
As a developer, I want to document my project so that future work can be easily continued	-Update guide based on our learnings/tasks	4	Neville	Done
As a developer, I want to multithread KNN so that the run time is reduced	-Multithread the fetching of data and the model	2	Neville	Done
As a developer, I want to run the non-NLP multithreaded models on the largest dataset	-Complete runs on all models: KNN, LSVC, RF, MLP, DT and give scores and time to run	2	Neville	Canceled
As a developer, I want to run all of our models on the largest dataset so that I can see the run time and accuracies	-Run all of the models and add the run time and accuracies to the table	2	Ziqian	Done
As an analyst, I want to create a presentation so that I can communicate our project's accomplishments	-Create a presentation in powerpoint -Add necessary information for complete	2	Olivia	Done

	communication			
As a student, I want to edit every section of the paper.	-Every team member adds edits to the paper	3	All	Done
As a student, I want to add/edit in a table of contents.	-Add a table of tables and table of figures	1	Ziqian	Done
As a student, I want to add machine learning backgrounds for the various techniques we used.	-Write background for all techniques used	2	Neville	Done
As a student, I want to add a machine learning methodology section for the library sklearn.	-Add a section on sklearn to the methodology section	1	Neville	Done
As a student, I want to add a technical learnings section.	-Add technical learnings to the assessment section	2	Luke	Done

Table 7.7.1 - User Stories in Sprint 7

Total Story Points	27

Table 7.7.2 - Total Story Points in Sprint 7

While finalizing our results, we made a lot of interesting discoveries. In sprint 4, we thought we had found the testing and training data were the same but upon closer examination, we found they were not as similar as we thought. As a result, we switched back to the original testing and training data to be authentic to Fidelity's procedures. In sprint 7 we also started running our code on the 9 million row dataset. We found that models with NLP had a significantly harder time running with this dataset, especially when multiple jobs were running at once. The team was concerned that run times resulting from NLP models that were run at the same time as other models would not be realistic. To resolve this, we created a schedule for using

control-M so that code that was being run for final results to be put in our results tables was not run alongside other code. Our final discovery was that multithreading is not worthwhile with scikit-learn. Testing results from multithreaded models came back with a greatly reduced accuracy, before multithreading accuracies were around 90% and, depending on the model, dropped to around 20%-40% after multithreading. While research does not say you cannot use multithreading with scikit-learn, multiprocessing appears to be the more successful use of parallelism in our case (Scikit-learn, n.d.). We decided to forgo multithreading, instead using single-threaded models and multiprocessing where applicable. Our final result of sprint 7 is that we will not be getting access to EMR and will rely on our results from our datasets in control-M for our final findings.

In our final retrospective, the team took time to reflect on what was done well in the past week. Since we did not have a sprint following sprint 7, we chose to reflect on what we would have done differently throughout the whole project for a final conclusion to our sprints. We also did not talk about what would be improved in the next sprint, as it was not relevant. As a whole, we were very happy with our work in the past week. We did a good job finalizing our project and drawing our final conclusions. When reflecting on our project as a whole, the team agreed that we could have been more organized at the beginning of the project in order to better execute the steps that needed to be taken to create our deliverables. For example, we felt a better job could have been done outlining the steps to creating our deliverable at the beginning of the term rather than having an idea of what needed to be done and adding user stories only once a step needed to be completed.

Due to our inability to gain access to EMR within our project timeline, we observe a significant operational risk. Fidelity requires those seeking to use EMR to make an access request. This limits the amount of people who have access to the data within EMR, giving a lower likelihood that data will be accidentally shared in a way that could be compromising for Fidelity. While this is a good risk control, in reality the process takes so long that progress is put at risk. To make this risk control better, there needs to be a balance between security and ease of access so that team progress is not inhibited.

7.8 Product Burndown Chart

To track our progress throughout the project timeline, the team has created a burndown chart. A burndown chart shows how much work is left, either in the form of hours or story points, and how much time is remaining to accomplish the work (Projectmanager.com, 2019). For our project, we tracked work using story points to align with Agile Scrum Methodology. The team also chose to track our time in the form of sprints rather than days. We were not finishing user stories everyday, so it felt redundant to track by day. Tracking by sprint gave us a more useful view of how much work we were accomplishing each sprint and how much was left to be done by the end of the project timeline.

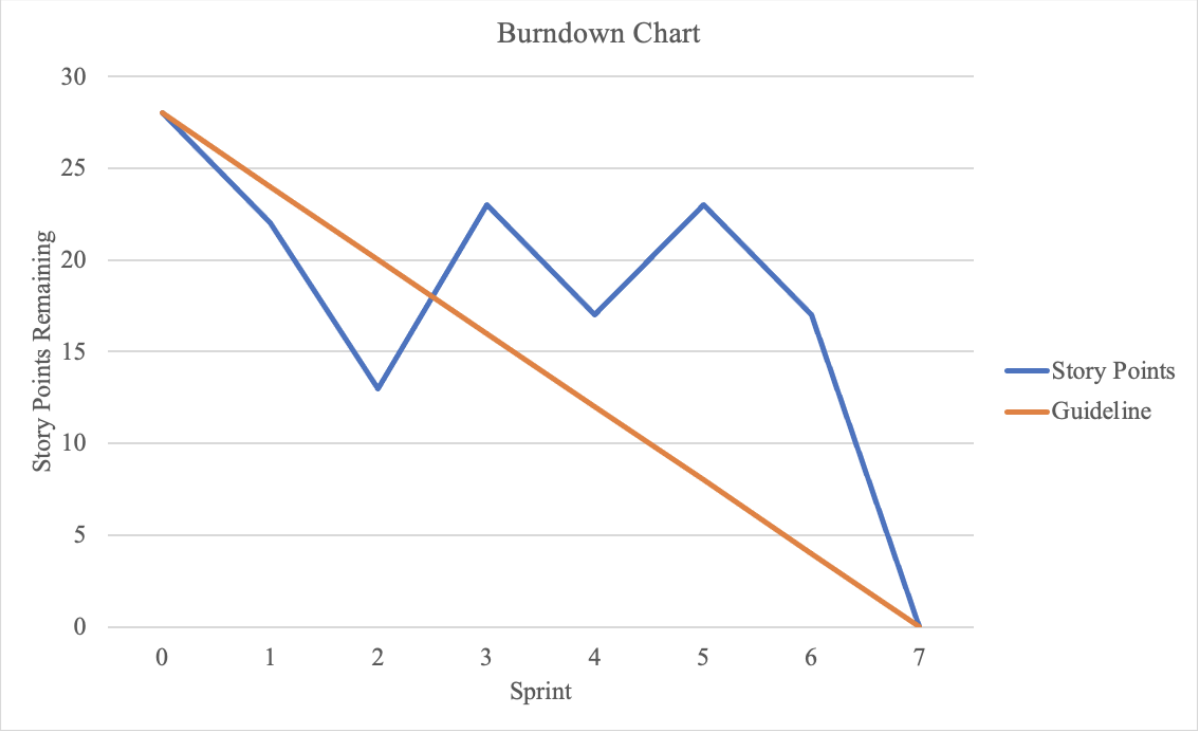

Figure 7.1 - Product Burndown Chart

The team chose to format the Y-axis using Fidelity's burndown chart format. Fidelity uses dates for the X-axis which we switched to the sprint number. As we learned more about our project, the team added more user stories. New user stories were added in sprints 3, 5, 6, and 7. A significant number of story points were added in sprints 3 and 5, which is why there is an

upward slope where one might expect a downward slope. More work was accomplished than user stories added in sprints 6 and 7, therefore the graph maintains a downward slope.

8. Business and Project Risk Management

8.1 Risk VS Reward

The potential risk that comes from our project not being successful comes from the run time and accuracy of the model. If the team's improvements to the run time and accuracy are insufficient for Fidelity's uses, more time will be spent with insufficient models and more resources will need to be used to make improvements.

The benefits of the successful completion of this project are extensive. First, a faster model would result in less time lost waiting for the runtime to carry out. A model that is able to execute quickly provides necessary procurement information more readily. Higher accuracy in the data returned by the model will also lead to better spend analysis. Spend analysis done on inaccurate data is worthless, whereas analysis on accurate data can be very valuable. Results from correct analysis can help Fidelity identify areas where costs may be reduced, compare pricing from various vendors, and track spending. From these benefits, we can assume that the improvements made by our project can help Fidelity indirectly and directly decrease costs. Indirectly by saving time and directly by identifying areas where costs might be reduced.

The largest benefit our project will yield for Fidelity is reducing the need for the spend classification correction form. Eventually, Fidelity would like to have a 100% accurate model to eliminate the need for classification correction. While our team was not able to reach this level of accuracy, we have made improvements which will result in less corrections needing to be made. Increasing the accuracy of the data used in Fidelity's procurement software also allows historical spend data to be leveraged better. If a user is going to predict the cost of a new purchase based on previous purchases, they need to be able to find the data necessary to do so. A misclassified purchase is unable to be leveraged if the user cannot find it. Misclassified data will also distort the amount of spending within a sector. If the user wants to see how much was spent on servers in the past year to negotiate a new price, it is necessary that the total amount spent on servers in the procurement software is accurate for the negotiation to be worthwhile (Swanson, 2021).

8.2 Risk Culture

Fidelity tends towards being risk-averse. For Fidelity, this is to protect their customers and their reputation. Customers investing through Fidelity or using one of Fidelity's many Fintech solutions often are required to share sensitive personal information. Alongside properly managing their customers' investments, Fidelity needs to ensure that all personal information is stored and handled securely. In order to maintain the trust of their customers, Fidelity's reputation needs to remain in good standing. If they choose to take a risk and it ends up unsuccessful, customers may question Fidelity's ability to manage risk and make appropriate investments based on the customer's risk preferences.

Our project team is also averse to risk, but not as intensely as Fidelity. We have taken on this attitude mainly from following our manager and Fidelity guidelines. The scale of our project timeline has also made us risk-averse. If we take a big risk and are unsuccessful, there is not much time to recover. Simultaneously, due to the short timeline, the team does not mind small risks for the sake of moving the project forward with the support of our manager.

Fidelity has taken several steps to manage risk within our project. The team has been paired with a team manager familiar with other Fidelity team members. The team manager is able to give our team guidance on Fidelity's software access protocols and connect us to people who can help. (How this mitigates risk). We have also been provided virtual machines to work on. Fidelity has required that all work be done on these machines to protect sensitive information. To help us understand what other actions we need to take to protect sensitive information all team members participated in information protection training. We also participated in model risk management training to help us understand how Fidelity defines risks in models and oversees risks from improper use of models. Lastly, the team participated in inclusion training to maintain Fidelity's workplace culture.

The team has also observed some risk controls around the data used in our project. These controls are in place to ensure data is entered completely and correctly. When accounting closes their books at the end of the month, their spend data goes through an approval process before being finalized. Only once the data is finalized is it entered into the procurement software. After

the data is entered and classification is complete, incorrect data is reclassified through the classification form. Combined, these processes help prevent incorrect data from being used in the procurement software.

8.3 Additional Risks

8.3.1 Operational Risks

Operational risks are caused by unproductive processes, systems, or events (AuditBoard, 2018). One operational risk observed over the course of the project was poor documentation on previous work and understanding of where work needed to be continued. This made it difficult to find required infrastructure for our project and where we needed to start work on the pre-existing machine learning models. The team, alongside our manager, ended up holding several meetings with Fidelity employees who could teach us about required infrastructure and those who we believed to have past experience with the project. Meetings to discover past work on the project were not successful, so we continued using the documentation we had and reading the code.

Meetings surrounding the infrastructure allowed us to gain access to software that we could access data and run code on. We were unfortunately never able to get access to EMR despite these meetings. Fortunately, we were able to use the software we had to develop work and come to conclusions. This exposed another operational risk - it takes a long time to receive final access approval for EMR. Speaking to our manager and squad leader, we discovered that it was not uncommon for teams to wait over a month to receive EMR access. The access approval process is important because it ensures that only employees that need to see the data that can be found through EMR have access. The more people who see this data, the more likely it is to be spread outside Fidelity or to people who might misconstrue the meaning of the data (Swanson, 2021). That being said, there should be a balance in between security and ease of access to maintain productivity in teams seeking EMR access.

Another operational risk that might have led to us having slow response time to our questions is that our team held a different sprint schedule than Fidelity. Fidelity's schedule operated over the course of two weeks from Wednesday to the second Tuesday. Meanwhile, our team's schedule was one week long from Sunday to Saturday. The difference in the schedule was not clearly communicated to all the Fidelity employees our team was dependent on, only our manager and Fidelity's scrum master. Other employees might not have understood our urgency to have problems resolved by Friday rather than at the beginning of the following week because of this. It was important for the team to clarify our project timeline as much as possible in order to mitigate this risk. Working with one of Fidelity's scrum masters also helped us learn the best ways to operate our epics in Jira without disturbing Fidelity's sprints.

8.3.2 Financial Risks

Financial risk is the loss of money from a venture (Hayes, 2021). With Fidelity's current model, inaccurate data may lead to poor decisions and financial loss. The procurement software this data is being used for helps employees identify vendors, compare costs, and find potential savings. If costs are misclassified, these activities might be performed incorrectly and indirectly lead to increased costs. Current mitigation for this risk is the classification correction form and the accounting closing the books at the end of the month. When accounting closes their books, the costs recorded are checked against the costs in the procurement software. Inconsistencies are investigated and misclassified data within the procurement software is edited using the classification correction form.

8.3.3 Reputational Risks

Reputation risk affects the standing of the company (Kenton, 2021). Within the procurement software, the spend data does not have any proof of acceptance or justification for the purchase. To someone unfamiliar with Fidelity's purchasing process, it might look as if employees are making any purchases they think necessary without approval from management (Weaver, 2021). If information from the procurement software was to be released without context, it might be a cause for concern amongst customers. In order to protect data from being

released, Fidelity restricts software access for employees that do not need it, employs internal and external cyber security, and holds employee training for data protection (Swanson, 2021). The external cyber security used by Fidelity seeks out and eliminates external threats. The internal cyber security tests how easy it is to break into Fidelity's servers and cloud by acting as someone trying to access Fidelity's data without permission. When tests are failed, improvements need to be made to security.

8.3.4 Innovation and Change Management Risks

Fidelity's risk culture tends to make employees very averse to innovation. As a result, their innovation and change management risks are intertwined. Innovation risk is the risk you take on when you make improvements to processes (Harvard Business Review, 2013). Change management risks are factors that might prevent a solution from being adopted (LaMarsh Global, 2020). Fidelity is very risk averse and tends to be cautious or hesitant when innovating. In order to stay relevant within their market, Fidelity must continuously improve within the company and for their customers. To continue innovating, Fidelity must find a way to do so in a controlled manner. Employees presenting possible innovation receive the advice to give proof of concept. Innovation appears more reliable when backed by tangible research and value. Providing evidence of why you think the innovation will work, steps to be taken to implement the innovation, and value added offers reassurance. Seeing the benefits of the innovation will help make people more willing to accept it.

8.3.5 Training Risks

Training risks are risks from not properly training employees (Boyle, 2015). An important part of employee training is giving access to or providing the necessary materials to prepare for a project. This is especially important when passing on a project that has previous work done. The team's project was the continuation of work done by a past intern. This intern had documented some of their work, but failed to include essential details such as the accuracy, run time overall and for the separate components of their code, and required software access. In order to find this information, the team started by asking employees with connections to the past

work if they could provide these details. We found that the only recorded work was documented in the past intern's guide. The team then proceeded by rerunning previous code to find the run time and accuracies and requesting access when it became clear we would need to use a particular software. Overall, this slowed our progress down a lot. For future work on the project, we will be recording all of this information in an updated guide similar to the past intern's documentation.

9. Assessment

9.1 Business Learnings

Working with Fidelity provided the team with many insights into teamwork and working at a large company. Within the company, we found the culture to be very risk averse. Upon further investigation, this appears to be standard for brokerage firms. An important aspect of Fidelity's business model is maintaining their customer's trust. Fidelity's (as with other brokerage firms) customers rely on them to make intelligent investing recommendations. If Fidelity is making risky decisions that are compromising the business, it reflects poorly on their ability to safely invest on behalf of their customers. Learning this helped the team assess our own risk levels and make decisions. In order to respect Fidelity's culture, we took on the same aversion to risk.

Fidelity's risk aversion often shows in their operations. Not only are all employees required to go through training to reduce many kinds of risk, but they also operate under strict standards. The team first experienced this in Jira, where the scrum master holds strict procedures for starting, working within, and ending every sprint. If these procedures are not followed, actions are flagged and reported in the Agile Data Quality program through PowerBI. The scrum master then takes responsibility to correct these actions. Fidelity also holds procedures for their workflow with internal projects. While it would seem most obvious that the developer would be able to directly communicate with their customer when developing internal Fidelity products, an intermediary is always involved. For example, if a team of data analysts were developing a product for accounting they would not directly communicate with their customers within accounting. Instead, they would communicate with a business analyst who acts as a bridge or translator between the data analysts and accounting. Once we learned this, the team was able to make more sense of our own communication flow.

Communication was something our team struggled with throughout the project, both internally and with Fidelity. We did not start our project with access to Fidelity's Jira and as a result had to find ways to temporarily store our sprint information in a safe way. We ended up

doing this through a Google doc so that the whole team could easily access the information and used very broad and unspecific terminology to protect Fidelity. This method resulted in our scrum meetings being disorganized and task focused rather than user story focused. We often did not know how what we had worked on impacted progress on the user stories and a lot of time was spent having discussions when not necessarily appropriate. When the team was able to start working on Fidelity's Jira, it took some adjustment to better organize our scrum meetings. Eventually, centering our scrum around the user stories and holding discussion at the end of scrum made meetings much shorter and led to more effective progress updates.

The team also believes that working remotely made it really hard to properly communicate. Having to communicate with our Fidelity coworkers through email and teams made asking simple questions take longer. We also think it made setting up meetings harder because rather than being able to hold short, informal meetings, every meeting ending up being scheduled up to a week in advance and an hour long. Within our team, we noticed that discussion tended to stagnate over Zoom and in person discussions were more fruitful. We often accidentally talked over each other when Zoom was lagging, which was distracting and made us lose focus. Team members living in Worcester eventually started meeting on campus to work together. This helped us make a lot of progress and improve our workflow.

The use of Agile development has also benefited the workflow of our project. While there was a learning curve and it took a few weeks to gain access to Fidelity's Jira, once we learned how to effectively use Agile we found it very functional. The format of the user stories helped us think about the purpose behind our actions and the intended results for every task. Holding daily scrums helped the team keep track of what work had been finished, future work, and blockers to our project. Scrum also presented an opportunity for the team to ask our project manager questions and have discussions. As the sprints progressed we learned how to more effectively organize our scrum meetings. In the beginning, each person presented what they had done, what they were going to do that day, and their concerns. Within each presentation, topic tasks were given in random order and discussion was held at random points. We eventually realized scrum would be much more efficient if we centered our presentation around user stories. From then on each team member would go through each user story in order and present the

status. At the end of scrum, the team would then ask questions, make general announcements, and hold discussions.

9.2 Technical Learnings

Working on this project taught the team many technical skills, ranging from software to libraries and machine learning models. We were introduced to the following new software: Snowflake, WinSCP, and Control-M. Snowflake is what Fidelity used for storing their procurement data, and we learned how to navigate it via SQL commands. WinSCP, a file transferring software, was used to move our code from our local machine to a location where the code can be executed. Control-M was the primary software we all learned, as it was what executed our scripts. We learned how to order jobs, execute them, and interpret the log files from them. Control-M was an important piece of software to learn, as Snowflake did not work when running code on our local machines.

All of the code we wrote was in Python, which the team was familiar with through past computer and data science classes at WPI. That said, a lot of the libraries used by the prior interns that worked on the project and that we ended up using were new to us. Specifically, sklearn was a library that we learned more about. Sklearn was mainly used for creating, testing, fitting, and scoring our machine learning models, making it an integral library to master. We also gained insights into how to multithread in Python, even if the code did not run well on Control-M. Other libraries that were new to us that we learned about include the pickle library for creating pickle files, Panda dataframes, and NLTK (Natural Language Toolkit), which had data science tools like stop words and lemmatization.

The team also learned about machine learning models that we later implemented or improved via code. The models we researched to try and implement were KNN, decision trees, MLP, and Gaussian process. Through that research we learned about how the models worked, their different parameters, and how to connect them to sklearn libraries in Python. We also familiarized ourselves with Linear SVC and random forests, as those were the two models the prior interns used and we wanted to understand any possible optimizations to either model.

Finally, we learned about natural language processing and understanding the stages of preprocessing it so we could better understand the prior interns' code and optimize it.

Overall, we got very well-acquainted with the software stack of Fidelity's procurement team. We got to learn not only about the various technologies, but also how to be software engineers and data scientists for a large company. Editing and interpreting pre-existing code was a learning experience for the team, as most class projects were from the ground-up whereas this real-world project required us to build upon the models and scripts of previous interns.

9.3 Takeaways

The team had several takeaways that would benefit future projects. First, is to discover software requirements before the project starts and request access where necessary. Throughout the project, we lost a lot of time waiting to learn how to gain access to necessary software. If this had been worked out before the project started, much more could have been accomplished. To avoid wasting time, the team found it important to clarify the project timeline and sprint schedule if different than the company's. Our project was much shorter than a classic internship but we found many employees we worked with did not know this. It is important that everyone involved in the project knows work must move quickly in order to resolve issues within the timeline. Employees at the company must also know your sprint schedule in order to help you accomplish tasks within your sprint. Otherwise they might assume that you are working on the same schedule. The team also found that working in person helped facilitate discussion and improved work efficiency. Adding new stories immediately when new tasks were started was also important to increasing work efficiency. New tasks often presented themselves in the middle of the week. It was much more convenient to add user stories for these tasks to the backlog immediately rather than waiting until the next sprint planning meeting. Adding the user stories immediately helped us remember all aspects of the task and keep our sprint planning meetings shorter because we did not need to add as many stories.

10. Future Work

During the term, our team focused on improving the existing machine learning models and introducing new models to classify Fidelity's spend data. Unfortunately due to time constraints and other roadblocks we were unable to reach the second part of our project - to improve the spend classification correction form. To prepare the future teams to work on this we will briefly outline some additional tasks we believe will lead to improvements. In addition to this section of our paper, we have created a Fidelity standard document that is more of a technical guide.

The second task focuses on adding features to a pre-existing classification correction form that is built with technologies such as Angular and NodeJS. The current form only allows users to edit the level 3 commodity classification but would be more useful with additional editing selections. The level 3 commodity is defined by a set of selection criteria. If one criterion is wrong, the result is a data point that has been classified to the wrong level 3 commodity. In this case, just editing the commodity does not resolve the issue. The ability to edit selection criteria will need to be added to the correction form in order to remedy this.

In addition to improving the spend classification form, the models used to classify spend data could be improved upon. Due to the time limitations of the term and the team's lack of knowledge in the subject, we did not explore deep learning as much as we would have liked to. However, from our limited research, it seems clear to us that deep learning is a highly accurate form of machine learning. In the future, a deep learning model could provide greater accuracy without needing to be retrained very often. During the term, the team implemented a multilayer perceptron. However, we could not run it in a reasonable timeframe because it would take over a full day to train on a subset of data. A future team could improve upon this model and explore methods to decrease the runtime, as the multilayer perceptron we had was very accurate (about 98%). There is also an existing codebase for tuning the multilayer perceptron, however only a select few of the parameters were tuned; future teams could continue this process as well.

A final note is not related to a technical suggestion, but is an organizational one. The process described in previous sections on how the code was executed was a hindrance to the

team. It provided a challenge to communicate which files held what code and if certain models were up to date. In standard coding environments, it is common practice to use some sort of source control for the team, and we recommend doing this. A common source control platform, Github, does have an API for control-m automation. Control-m “job flows and related configuration objects are built in JSON and managed together with other application artifacts in any source code management solution, such as GIT” (Schatz Yechezkel, 2021). Having a source control platform such as Github allows for advantages that are organizational in nature and will simplify the workflow between future teams.

11. Conclusion

Fidelity manages billions of dollars for purchasing equipment, technology, and other assets to run their business. To better understand their budgeting and classify their extensive spending tables, they have utilized machine learning models. These models, however, have performed slowly and lacked optimization, which is where our team stepped in. Our first task was to optimize the machine learning process through developing new models, optimizing our new models and their pre-existing ones, and leveraging multi-threading and other time-optimizing methods to speed up the process while increasing accuracy.

The first few weeks were spent setting up and familiarizing ourselves with the technologies in Fidelity's virtual desktop and getting concrete project-direction from our managers. Then, we began researching models and testing them on the small dataset. After picking the ones that worked, we studied different parameters and tried adding NLP to the models. After that, we used a much larger dataset and did further fine-tuning on the models and selected the most accurate and fastest models to be ready for EMR testing. We also wrote multithreading code for the models we selected and fixed the prior interns' code.

At the end of the project, we were able to draw several conclusions from the first focus despite not getting EMR access. The first conclusion was, after testing several different models and parameters with the 10,000 row table on Control-M, we deducted that this method was not valid for getting data. This is because the training scores were extremely high accuracy (some were even 100%), which essentially means overfitting is occurring. We concluded that the overfitting was occurring due to the smaller data set, so we moved on to testing with the 5 million row table. However, we discovered that the create model and execute model workflow did not work for the 5 million row table. We suspect this is due to Control-M not being suitable for large datasets and thus running out of system resources. Our solution was to then use a different method to compare Fidelity's older models to the ones we made: sklearn's train_test_split function.

We developed a different file for testing the 5 million row table which involves using sklearn's train_test_split function after preprocessing, then performing training and testing at the

same time. Table 11.1 shows the runtimes and accuracies we got from running four models: KNN, Decision Tree, and the two models that Fidelity used prior to our arrival, being RF and Linear SVC. The table reveals that KNN and Decision Trees were both faster than Linear SVC. We also added an optimization to Random Forest by adding the parameter n_jobs=-1 to it, which utilizes multiprocessing. The accuracies, however, are not necessarily accurate due to this method differing from the create/execute model method that Fidelity runs on the EMR, but it is sufficient for concluding which model is faster since they comparatively run faster.

Models	Accuracy	Run Time (HH:MM:SS)
K Nearest Neighbour	98.20%	0:45:04
Decision Tree	98.58%	0:01:34
Random Forest	98.53%	0:21:44
Linear SVC	97.52%	1:18:31
Run Time = **Processing Time** + **Training Time** + **Testing Time**. (Data Fetching Time is very similar of all runs therefore has been removed from the runtime calculation of this table)		
Testing Method: **train_test_split()**		
Snowflake Table: **"FCT_PROCUREMENT_SPEND_STG_BKP"**		

Table 11.1 - Summarized ML Model Performance on 5 Million Rows of Spend Data

We also fixed the multithreading code provided by Fidelity and managed to execute it, however, the results showed a decrease in accuracy compared to running the models without multithreading. We believe this problem is due to either incorrect code or an issue with Control-M. We thus concluded that it was better to proceed with testing the multithreading code on the EMR or simply use multiprocessing, as it is a built-in tool of sklearn. The biggest takeaway from our project was that using decision trees resulted in roughly the same accuracy in a fraction of time computational time, which thus saves Fidelity valuable time.

We were unable to get to task two, which was to expand the spend correction form via AngularJS, due to the limited amount of time we had on this project.

Throughout this project, we also learned a lot about Fidelity and their process of software development and agile scrum. We also got to learn and expand our knowledge of various technologies, including Control-M, WinSCP, and Python. All in all, we were able to provide

Fidelity with different machine learning models that can increase the efficiency and accuracy of their monthly procurement process.

12. References

Accelerated Insight. “What Is Spend Data Classification?” Accessed November 3, 2021. https://www.accelerated-insight.com/spend-analysis/what-is-spend-data-classification.

Apoorva Srivastava, Sukriti Bhardwaj, and Shipra Saraswat. 2017. "SCRUM Model for Agile Methodology ." . https://ieeexplore.ieee.org/document/8229928.

Atlassian. “Jira Work Management: A Friendly and Powerful Way to Work.” Accessed November 10, 2021. https://www.atlassian.com/software/jira/work-management.

Atlassian. “User Stories | Examples and Template.” Atlassian. Accessed November 10, 2021. https://www.atlassian.com/agile/project-management/user-stories.

AuditBoard. “What Is Operational Risk Management? The Overview.” Accessed November 22, 2021. https://www.auditboard.com/blog/operational-risk-management/.

BMC Software. “Job scheduling and Workload Automation” Accessed November 18, 2021. https://www.bmc.com/it-solutions/job-scheduling-workload-automation.html.

Brownlee, Jason. “One-vs-Rest and One-vs-One for Multi-Class Classification.” *Machine Learning Mastery* (blog), April 12, 2020. https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/.

Brownlee, Jason. “Crash Course On Multi-Layer Perceptron Neural Networks.” *Machine Learning Mastery* (blog), May 16, 2016. https://machinelearningmastery.com/neural-networks-crash-course/.

Corporate Technology Group. “Spend Classification & Forecasting WPI Intern Project” Boston: Fidelity Investments, September 23, 2021

DataCadamia. "Statistics - Residual Sum of Squares (RSS) = Squared Loss ?", last modified May 04, https://datacadamia.com/data_mining/rss#about.

DeepAI. "What is a Hyperplane? .", last modified Feb 19, https://deepai.org/machine-learning-glossary-and-terms/hyperplane.

Expert.ai. “What Is Machine Learning? A Definition.,” May 5, 2020. https://www.expert.ai/blog/machine-learning-definition/.

Fidelity. “About Fidelity - Our Company.” Accessed November 3, 2021. https://www.fidelity.com/about-fidelity/our-company.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, eds. 2021. *An Introduction to Statistical Learning*. 2nd ed. https://hastie.su.domains/ISLR2/ISLRv2_website.pdf.

Harvard Business Review. “Innovation Risk: How to Make Smarter Decisions.” *Harvard Business Review*, April 1, 2013. https://hbr.org/2013/04/innovation-risk-how-to-make-smarter-decisions.

Hayes, Adam. “Financial Risk: The Art of Assessing If a Company Is a Good Buy.” Investopedia, March 24, 2021. https://www.investopedia.com/terms/f/financialrisk.asp.

IBM. “What Is Natural Language Processing?,” July 2, 2020. https://www.ibm.com/cloud/learn/natural-language-processing.

IBM. “What Is Supervised Learning?” Accessed December 1, 2021. https://www.ibm.com/cloud/learn/supervised-learning.

IBM Cloud Education. "What is Bagging?", last modified 11 May, https://www.ibm.com/cloud/learn/bagging.

Jetbrains. “Feature - Pycharm” Accessed November 14, 2021. https://www.jetbrains.com/pycharm/features.

Kenton, Will. “Reputational Risk.” Investopedia, October 10, 2021. https://www.investopedia.com/terms/r/reputational-risk.asp.

LaMarsh Global. “Guide to Change Management Risk Assessment,” July 6, 2020. https://insights.lamarsh.com/guide-to-change-management-risk-assessment.

M.W Gardner and S.R Dorling. 1998. "Artificial Neural Networks (the Multilayer Perceptron)—a Review of Applications in the Atmospheric Sciences." . https://www.sciencedirect.com/science/article/pii/S1352231097004470?casa_token=M5sHEcqhqq0AAAAA:L1k9975EZyysfW-Eu3P1oQRHj3dzdvSdUAcFEwQ0jA-SX1wxbK4BeyHeJSnvAW5YjHsq7XTMyA.

Martin, Matthew. “Functional Requirements vs Non Functional Requirements: Differences.” Accessed November 10, 2021. https://www.guru99.com/functional-vs-non-functional-requirements.html.

Microsoft. “Documentation for Visual Studio Code” Accessed November 14, 2021. https://code.visualstudio.com/docs.

Microsoft. “What is Power BI: Microsoft Power BI” Accessed November 18, 2021. https://powerbi.microsoft.com/en-us/what-is-power-bi/

NewsBank. “Automatic Spend Classification Helps Overcome the Challenge of Unreliable Spend Data.” ICT Monitor Worldwide (Amman, Jordan), November 28, 2019. https://infoweb.newsbank.com/apps/news/user/login?destination=document-view%3Fp%3DAWNB%26docref%3Dnews/1777F60886FC40F8.

ProductPlan. “Epic.” Accessed November 18, 2021. https://www.productplan.com/glossary/epic/.

ProjectManager.com. “Burndown Chart: What Is It & How Do I Use It?,” February 13, 2019. https://www.projectmanager.com/blog/burndown-chart-what-is-it.

Rehkopf, Max. “User Stories: Examples and Template.” Atlassian, n.d. https://www.atlassian.com/agile/project-management/user-stories.

REHKOPF MAX. "Sprints." . https://www.atlassian.com/agile/scrum/sprints.

REHKOPF MAX. "User Stories with Examples and a Template ." https://www.atlassian.com/agile/project-management/user-stories#:~:text=A%20user%20story%20is%20the,from%20the%20software%20user%27s%20perspective.&text=Stories%20fit%20neatly%20into%20agile,the%20duration%20of%20the%20sprint.

Schwaber, Ken. *Agile Project Management with Scrum*. Microsoft Press, 2004. https://www.agileleanhouse.com/lib/lib/People/KenSchwaber/Agile%20Project%20Management%20With%20Scrum%20-www.itworkss.com.pdf.

scikit-learn. “Glossary of Common Terms and API Elements.” Accessed November 10, 2021. https://scikit-learn/stable/glossary.html.

scikit-learn developers. "Decision Trees.", https://scikit-learn.org/stable/modules/tree.html#tree.

———. "Nearest Neighbors.", https://scikit-learn.org/stable/modules/neighbors.html#classification.

scikit-learn. “8.3. Parallelism, Resource Management, and Configuration.” Accessed December 9, 2021. https://scikit-learn/stable/computing/parallelism.html.

scikit-learn. “Sklearn.Ensemble.RandomForestClassifier.” Accessed November 10, 2021. https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

scikit-learn. “1.4. Support Vector Machines.” Accessed November 10, 2021. https://scikit-learn/stable/modules/svm.html.

Snowflake. “Snowflake's Cloud Data Platform: One Platform for All Your Data” Accessed November 18, 2021. https://www.snowflake.com/cloud-data-platform/.

“Spend Analysis Delivers Big Benefits.” APQC, n.d. https://www.apqc.org/sites/default/files/files/Spend%20Analysis%20Delivers%20Big%20Benefits.pdf.

Srivastava, Apoorva, Sukriti Bhardwaj, and Shipra Saraswat. “SCRUM Model for Agile Methodology.” In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 864–69, 2017. https://doi.org/10.1109/CCAA.2017.8229928.

Swanson, Erik. Interview with Erik Swanson on risks, data security, and benefits relevant to our project, November 30, 2021.

Weaver, Daniel. Interview with Daniel Weaver on Procurement Software, November 19, 2021.

West Dave. "Sprint Planning." . https://www.atlassian.com/agile/scrum/sprint-planning#:~:text=Sprint%20planning%20is%20an%20event%20in%20scrum%20that%20kicks%20off,that%20work%20will%20be%20achieved.&text=The%20What%20%E2%80%93%20The%20product%20owner,items%20contribute%20to%20that%20goal.

WinSCP. “Introducing WinSCP” Accessed November 18, 2021. https://WinSCP.net/eng/docs/introduction

Yiu, Tony. “Understanding Random Forest.” Medium, September 29, 2021. https://towardsdatascience.com/understanding-random-forest-58381e0602d2.