Fidelity: AI Based Anomaly Detection

Project Team:

Aruzhan Koshkarova (akoshkarova@wpi.edu)

Austin Zhou (azhou@wpi.edu)

Mitchell Sirois (mssirosis@wpi.edu)

Nathan Kumar (nmkumar@wpi.edu)

Project Advisors

Professor Robert Sarnie

WPI Business School

Professor Marcel Y. Blais

Department of Mathematical Sciences

Professor Wilson Wong

Department of Computer Science

Abstract

This project's goal is to create a machine learning model that predicts data anomalies within Fidelity's assets and flows dataset. Anomalies occur in three models that are utilized in Snowflake: the TA model (customer holding), FA model (Fidelity fund holding), and Look Through model (consolidated holdings). The models reflect day to day changes in account and fund balances. Often, there are discrepancies in the data due to the process of patching the models together. Fidelity employs a team that is tasked with ensuring data integrity, which includes identifying errors in the models. In the present, errors are time consuming to find and fix. The MQP team proposed AI detection model solutions to quickly and accurately identify anomalies in the TA/FA data consolidation process.

Executive Summary

This MQP report centers around a solution-based approach to detecting anomalies in Fidelity's assets and flows data. The report details the step-by-step process that the team took, starting from background research, to understanding the data set, as well as testing and scaling viable solutions.

As an investment company with over 40 million customers, Fidelity deals with millions of fund transactions on a daily basis. The company invests for customers in three main realms: workplace investing, institutional investing, and retail investing. For each realm, Fidelity holds large funds that customers can buy pieces of for their own accounts. Fidelity organizes smaller funds to roll up into larger funds; in total there are twenty one levels of funds.

There are three models that track the flow of assets between accounts and funds. The Transaction Account (TA) table tracks customer account holdings. Fidelity uses a "TA bridge" to consolidate changes in the customer account balances. In theory, these bridges should accurately produce the correct final account balance. The Fund Account (FA) table tracks Fidelity fund amounts. There is currently no bridge for the FA table. A third table, the Look Through table is a result of patching the TA and FA tables together. For a variety of reasons, the data in the finalized Look Through Model may not accurately reflect the final account and fund balances. In order to resolve this issue, Fidelity deploys a team of associates to go through and look for errors in the data. This process can be quite time consuming and tedious.

The team first learned Fidelity's data flow and worked on the TA data to find anomalies within the managed assets. The team then produced and tested various SQL queries to validate that transactions and daily balance were being properly recorded. Furthermore, we worked in python to detect anomalies on a fund account level.

The team also researched various ML algorithms to identify anomalies that occur within the data as possible next steps. These techniques are described throughout the paper and offer solutions on how to find anomalies, find the cause of anomalies, fix anomalies, and eventually predict when they will happen. Being able to find the cause of the anomalies would allow for Fidelity to take preventative measures to reduce the number of anomalies that happen in the future.

Acknowledgements

The team would like to thank all of our professors and Fidelity sponsors for their support throughout the duration of this project. Our advisors, Professor Robert Sarnie, Professor Marcel Blais, and Professor Wilson Wong provided us constant guidance, encouragement and support that we needed to stay on track, and produce a final project that we were happy with. On the Fidelity side of things, every associate that we worked with was helpful in acclimating us to the workspace and did a great job teaching and accommodating us throughout the duration of the project. Specifically, we would like to thank our project head-sponsor, David Wolf, for challenging us and giving us the freedom to make this project our own. Our technical sponsors, Abhisheg Salvaraj, Mark McGough, Azarudeen Ahamed, and Shiny Abitha Judith gave us great insights and guidance to help steer our project in the right direction. Finally, our project liaison, Rick Snyder, was there for us throughout the duration of the project. Anytime we needed help or had any questions, he would help answer or find someone to answer those questions for us.

Table of Contents

List of Figures

List of Tables

Authorship

Section	Main Author(s)	Main Editor(s)
Cover Page	A. Zhou	
Abstract	N. Kumar	A. Zhou M. Sirois
Executive Summary	M. Sirois	A. Zhou N. Kumar
Acknowledgements	A. Zhou	N. Kumar
1. Introduction	A. Zhou	M. Sirois N. Kumar
2. Research		
2.1 Project Company Background	A. Zhou A. Koshkarova M. Sirois N. Kumar	A. Zhou
2.2 AI Anomaly Detection	M. Sirois A. Koshkarova N. Kumar	N. Kumar
2.3 Entity Relationship Diagram	A. Koshkarova N. Kumar	A. Zhou
2.4 Snowflake	A. Koshkarova	N. Kumar
3. Methodology		
3.1 Agile Methodology - Agile Scrum	A. Zhou M. Sirois	N. Kumar
3.2 Scaled Agile	A. Zhou M. Sirois	N. Kumar
3.3 Agile at Fidelity	A. Zhou M. Sirois	N. Kumar
3.4 Data and Computer Science Techniques	A. Koshkarova N. Kumar	M. Sirois

Section	Main Author(s)	Main Editor(s)
4. Software Development Environment		
4.1 Background/Access Software	N. Kumar	A. Koshkarova A. Zhou
4.2 Agile Software	A. Zhou M. Sirois N. Kumar	A. Koshkarova A. Zhou
4.3 Data Access and Analysis	A. Koshkarova N. Kumar	M. Sirois
5. Software Requirements		
5.1 Requirements Gathering	N. Kumar M. Sirois A. Koshkarova	A. Koshkarova
5.2 Functional Requirements	N. Kumar M. Sirois A. Koshkarova	A. Zhou
5.3 Epics and User Stories	A. Zhou M. Sirois A. Koshkarova	A. Zhou N. Kumar
6. Business and Project Risk Management		
6.1 Business Risk	A. Zhou	M. Sirois N. Kumar A. Koshkarova
6.2 Project Risk	A. Zhou	M. Sirois N. Kumar
7. Design		
7.1 Asset Bridge	A. Koshkarova A. Zhou M. Sirois N. Kumar	
7.2 Entity Relationship Diagram	A. Koshkarova A. Zhou M. Sirois	

	N. Kumar	
8. Software Development		
8.1 Scrum Details	A. Zhou	M. Sirois
8.2-8.8 Sprint X	N. Kumar M. Sirois	A. Koshkarova
8.9 Burndown Chart	M. Sirois	N. Kumar
9. Assessment		
9.1 Goals Reached	M. Sirois	
9.2 Learning Experience	M. Sirois N. Kumar	A. Koshkarova
10. Future Work		
10.1 TA Data	A. Zhou N. Kumar	M. Sirois
10.2 Asset Bridge	A. Zhou N. Kumar	A. Koshkarova M. Sirois
10.3 Application	A. Koshkarova N. Kumar	M. Sirois
11. Conclusion	M. Sirois	A. Zhou N. Kumar
12. References	A. Koshkarova M. Sirois	
13. Appendix		
13.1 Appendix A - Python Code	A. Koshkarova N. Kumar	M. Sirois
13.2 Appendix B - SQL Query for TA Data	A. Zhou	M. Sirois

1. Introduction

In this project, the team was tasked with creating and implementing a system that was capable of identifying anomalies within Fidelity's assets and holdings data. The scope of the project was primarily focused on three models that each represented a stage in the consolidation process.

The TA model is a record of customer holdings, including both the current state of all customer accounts, and also a recorded history of transactions related to these accounts. The FA model is similar; however, instead of customer accounts and holdings, it is focused on Fidelity-owned and managed funds. These two models intersect to create the Look Through Model. The Look Through Model is the overall structure of Fidelity's holdings, seen through 21 levels of look through. It begins at level one, which is the broadest view of Fidelity's funds, and descends through the levels until reaching customer facing products at level 21. Bridges are connections between accounts and their states (such as an account's beginning of day and ending of day balance) and contain information about completed transactions (by observing all inflows and outflows for a specific account, for example). Such a large and interconnected system proves laborious to search and troubleshoot when errors occur.

Currently, anomalies are detected when the reported ending day balance has deviated from the ideal value of the beginning of day balance plus inflows minus outflows. A team of associates is tasked with manually investigating the data to find the origin of the anomaly and adjusting any sources of error. This process is tedious and time consuming, leading to the requirement that a previous month's transactions cannot be approved until seven business days after the end of the month. Fidelity wishes to optimize this manual process using machine learning to free up its associates for other activities and thus has tasked our team with the goal of creating an efficient method for identifying these anomalies as they occur. To do this, the team set goals to develop anomaly detection models to quickly identify these anomalies and alert the analyst team of the errors and their origins.

2. Research

2.1 Project Company Background

Fidelity Investments is a large financial services company headquartered in Boston, Massachusetts . As of Quarter 3, 2022, the company has $3.6 trillion USD Assets Under Management (AUM), $9.6 trillion USD Assets Under Administration (AUA), and employs over 57,000 associates, with offices in nine countries (Fidelity-*Quarterly Updates*, 2022).

Customers

Fidelity Investments currently provides financial services with more than 40 million people, including individual investors, employers, advisors and institutions, charitable donors and innovators. The services they provide for individual investors are “financial planning, advice, and educational resources… including retirement planning, wealth management, brokerage services, college savings and more” (Mission, 2022). Along with individual customers, they also manage employee programs for over 23,000 businesses. The programs include retirement savings, health and welfare, and stock investing advice. Fidelity is also involved in supporting advisory firms, with over 3,600 managed accounts. In a volatile and changing industry, Fidelity offers solutions backed by technology and industry leading customer experience. Lastly, Fidelity Charitable offers customers the ability to donate to various causes in a simple and effective way.

History

The Fidelity Fund, founded in 1930, was taken over in 1943 by Edward C. Johnson. For the next 30 years, Mr. Johnson transformed the fund into Management and Research Company with focus on technology research, innovation and active investment management.

At the end of the 1960s, Fidelity started expanding internationally, opening its first research office in Japan and later expanding to the United Kingdom. In the 1970s Edward C. Johnson III diversified the business by adopting innovative services and technology to raise the customer service to the world-class standard.

Fidelity LLC was the first firm to offer 24-hour mutual fund yield in 1979, as well as the first mutual fund company to go online in 1995, which allowed for the online trading of mutual

funds. The private ownership of Fidelity until the 1990s allowed for protection from short-term market fluctuations as company investors could trust in the integrity of the fund.

In October of 2007 FMR Corp. refiled with the Securities and Exchange Commission (SEC) to transition into a limited liability corporation. This act of corporate restructuring led to a shifting of tax burdens from the company itself onto individual investors. Fidelity's proportions of ownership were unchanged by this decision, and the company continues to be privately held (Weiss, 2007).

Today Fidelity is at the forefront of innovating new financial products and services to customers. Most recently, Fidelity Spire and the establishment of Fidelity Digital Assets has allowed Fidelity to expand into emerging FinTech markets like retail investing and cryptocurrency.

Current Events and FinTech

As of September 2022, Fidelity averages around 3.1 million daily trades. In the past year, Fidelity experienced a growth of 11% in discretionary assets as well as a 13% increase in retail accounts (Fidelity-*Quarterly Updates*, 2022). Heavily involved in cryptocurrency, blockchain, and artificial intelligence, the company spends around $2.5 billion USD per year on its technology (Rooney, 2018). In 2017, Fidelity hosted the largest blockchain conference in the United States, to discuss various benefits and products. Many of these products are tested in the startup environment called Fidelity Labs. In 2017, Fidelity Labs launched a student debt program, offering and suggesting repayment options to pay off loans. The following year, Fidelity Digital Assets was born where they offer services to companies and institutions that invest in Bitcoin.

Competition

As Fidelity exists within the financial services and wealth/asset management sectors, they have a large pool of competition. These competitors range from large multinational wealth management companies, such as BlackRock and JP Morgan, to smaller, more local investment companies. Within the past couple of years, Fidelity has looked to take advantage of the simultaneous rise of both personal investing and crypto currency that has increased the population of retail investors in the market.

Culture

Fidelity's mission is to "strengthen the well-being of its clients" by "making financial expertise broadly accessible and effective in helping people live the lives they want" (Mission, 2022). They do so by operating under three core values: integrity, honestly, and loyalty. The culture at Fidelity is focused on employee well-being. The logic being that if employees are content and motivated to work, then they will produce better results within the business and for clients. The company hosts social events in the office, promotes DEI and employee resource groups, and implements learning days to build community within the workspace. Because Fidelity likes to invest in its employees, it is common to see associates make horizontal jumps across the company's business units to new roles that they may not have experience in. The company would rather train and develop their employees than constantly look to hire.

From a leadership perspective, Fidelity emphasizes a servant leadership approach. Managers and squad leaders know their squad members' work and workload well. For the most part, managers will meet individually with squad members weekly to check-in and provide any advice on work being done. Senior level managers also make the effort to interact with and talk to squad members whenever they have the time. This promotes stronger relationships between managers and squad members, which allows for greater interconnectivity and communication within the business.

As a private company, Fidelity is not beholden to shareholders. In this sense, the company's leadership team has the flexibility to take more risk and make more long term investments than a publicly traded company. For example, since 2008, there has been a great shift for Fidelity's allocation of resources. The company has transitioned from being a purely financial company to a tech/financial company.

2.2 AI Anomaly Detection

An anomaly (outlier) can be described as a data point in a dataset that is significantly different from other data or observations. This can happen for several reasons: outliers may indicate data that is incorrect, or an experiment may not have been done correctly. Outliers could be caused by a random change, or it could indicate something scientifically interesting (Alla & Adari, 2019).

The anomaly detection problem for time series is identifying outlier data points relative to some norm or usual signal. There are a few techniques that analysts can employ to identify different anomalies in data. It starts with a basic statistical decomposition and can work up to autoencoders:

2.2.1 STL Decomposition

Seasonal trend decomposition using LOESS (STL) is a robust time series decomposition technique that is often used in economic and environmental analysis. The STL method uses computational regression models to decompose a time series into trend, seasonality, and statistics components (Cleveland & Terpenning, 1990).

The STL algorithm smoothes the time series with LOESS in two cycles; the inner loop iterates between seasonal and trend smoothing, and the outer loop minimizes the impact of outliers. During the inner loop, the seasonal component is calculated first and removed to calculate the trend component. The remainder is calculated by subtracting the seasonal and trend components from the time series.

These three components of STL analysis are related to raw time series as follows:

$$y_i = s_i + t_i + r_i$$

where y_i= the value of the time series at point i; s_i=the value of the seasonal component at point i; t_i=the value of the trend component at point i; r_i=the value of the remainder component at point i.

The seasonal component of the STL result shows a recurring temporal pattern in the data based on the selected seasonality. If there is a seasonal pattern, then it usually takes the form of an oscillating or wave pattern.The trend component is the second component that is calculated during the inner loop. The values for the seasonal component are subtracted from the raw data so that the seasonal variation is highlighted in the time series. Then, by applying LOESS, a smoothed trend line is created for the rest of the values.

2.2.2 Classification and Regression Decision Trees (CART)

This method uses numerous decision trees to identify a point on a time series chart that is an anomaly. It uses the Isolation Forest Algorithm that will detect whether or not a point in time

series is an outlier or not. The more data this model has, the better it can recognize variance (Scikit Learn).

This technique will involve using supervised machine learning to teach the model what is and what is not an anomaly. Because this is supervised machine learning technique, developers will have to manually feed the model anomalies. The algorithm has the capability to predict whether a certain point on a time series graph is an outlier or not. The algorithm will fit and train the data, returning -1 for an anomaly and 1 for good data. From here, developers need to train the model based on the amount of outliers that are present in the data, this is done with trial and error. Scikit-learn library can then be used to implement the algorithm.

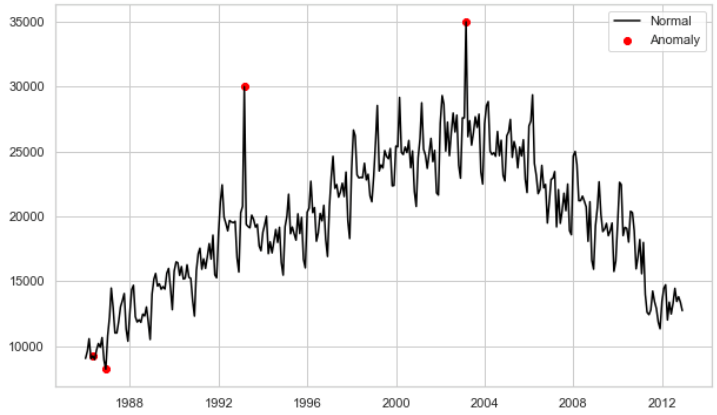


Figure 1: Results of Isolation Forest Algorithm (Bajaj, 2022)

The results show the data points that are considered anomalies. This could be a feasible option to detect anomalies in the assets and flows data.

The Isolation Forest (IF) is an example of an unsupervised algorithm that is able to detect outliers or anomalies in a dataset very quickly (Liu & Zhou, 2009).

The great thing about IF is that it can directly detect anomalies using isolation (how far a data point is from the rest of the data). This means that the algorithm can operate with linear time complexity, like other distance-related models such as K-nearest neighbors. The algorithm works based on the most obvious outlier attributes: there will only be a few deviations; their emissions will be different. Isolation Forest does this by introducing an ensemble of binary trees that recursively generate partitions by randomly choosing a feature and then randomly choosing a split value for the feature. The splitting process will continue until it has separated all data points from the rest of the samples. Since only one feature is selected from the instance for each tree,

we can say that the maximum depth of the decision tree is actually one. In fact, the base score of the Isolating Forest is actually an extremely random decision tree for various subsets of the data (Brunner et al., 2021).

In IF, developers recursively split each instance of the data by randomly choosing attribute q and split value p (within the minimum and maximum value of attribute q) until they are all completely isolated. The isolated forest will then provide a ranking that reflects the degree of anomaly of each data instance according to its path length.

2.2.3 Detection Through Forecasting

This method utilizes the creation and maintenance of a standardized forecast to detect when anomalies occur. This can be achieved through training a model through the use of trainings such as SARIMA or Auto Arima, or a simpler model can be created through a rolling average and standard deviation. This simpler method can be preferred however, as a more complex model will attempt to fit itself too closely to the data (Krishnan, 2019).

The ARIMA model works using a distributed latency model in which algorithms are used to predict the future based on values with latency.

The ARIMA model has subclasses of other models such as autoregressive (AR), moving average (MA), and moving average autoregressive (ARMA) models. For seasonal forecasting of time series, Box and Jenkins proposed a rather successful version of the ARIMA model, namely, its seasonal version - SARIMA. A serious limitation of these models is the assumed linear form of connected time series, which becomes inadequate in some practical situations. For this, some authors have proposed various nonlinear stochastic models. However, from the point of view of implementation, these models are not so simple and convenient, like ARIMA models (Nau).

The creation of an ARIMA model consists of four systematic steps (identification, evaluation, diagnostic verification, and application or prediction). The components of the series for removal were studied and established by the stl method - seasonal and trend decomposition using the LOESS ("STL") method.

2.3 Entity Relationship Diagram

An entity relationship diagram (ERD) helps show the relationship between objects in an information technology setting. Because the team was going to be dealing with large sets of data

with many tables joined together, we knew that an ERD would help organize and understand the flow of data through Fidelity's assets and flows system.

There are three types of models for ERDs: conceptual, logical, and physical (Biscobing, 2019). In a conceptual model, the purpose is to provide a high level overview of how data sets relate to each other. In a logical model, the diagram provides more detail on specific attributes and relationships among data points. A physical model is the most detailed type of ERD; it is essentially a blueprint of a logical model and contains all details needed to physically create the proposed information system.

ERDs consist of five main components: entities, attributes, relationships, actions, and connecting lines. Entities are the tables or objects that have data stored in them. Attributes are the individual properties and characteristics that make up entities. Primary keys are attributes that represent a unique attribute in an entity, while foreign keys can be assigned to multiple attributes. Relationships between entities are created by lines drawn from an attribute of one entity to an attribute of another entity. Actions describe how entities share information in the database. Connecting lines represent the relationship between identities. For our group's ERD, we focused on using cardinal notation, which defines relationships as one-to-one, one-to-many, and many-to-many.

2.4 Snowflake

Before the team gained access to the Snowflake data, we did some background research and training with Snowflake. Fidelity has shifted many of its databases (including the Assets and Flows data) over to Snowflake for its advantages in speed, collaboration, security and governance. Snowflake is a cloud based data platform that provides data storage and analytics services. Its data architecture consists of three layers: database storage, query processing, and cloud services. At a database level, Snowflake manages the organization, file size, structure, compression, metadata, and statistics of data that is stored (Snowflake). To process queries, Snowflake creates independent virtual warehouses. The benefit of these warehouses is that they share computing power with each other, resulting in more efficient query processing across a network. Snowflake offers a variety of management control services including: authentication, infrastructure management, metadata management, query parsing/optimization, and access

control. In training, the team learned how to access the warehouse and databases, use SQL queries to pull the data, and export data pulled from queries.

3. Methodology

3.1 Agile Methodology - Agile Scrum

The basis of Agile Scrum is the combination of agile philosophy mixed with the scrum framework. Agile is the idea of developing big projects in increments. Teams can more efficiently manage projects through breaking down projects into working stages by allowing for consistent collaboration at a steady pace. Scum is a specific type of agile methodology that is most beneficial for companies working in fast paced environments with complex projects. Agile scrum is designed to focus on efficiency and innovation by using specific roles, events, and tools (Peek, 2022).

3.1.1 Scrum Roles

The Scrum team consists of three different roles: the Scrum Master, the product owner, and Developers.

The Scrum Master serves as the leader for the group. They wear many hats to serve their team, Product Owners, and the organization. At a team level, their role includes: helping team members self-manage and work cross functionally, ensuring that increments meet high-value standards, removing any obstacles in the way of the team, and leading positive and productive Scrum meetings. Scrum Masters also work with Product Owners to ensure that product goals are defined and that the product backlogs are properly managed. In this sense, Scrum Masters serve as a bridge between group members and product owners for providing clarity and understanding in backlog items. For more complex projects, the Scrum Master also helps bring in and facilitate collaboration with outside stakeholders.

The Product Owner serves as a representative for stakeholders, who are typically customers. They are tasked with developing and explicitly communicating product goals, recording changes to the product, and creating and administering a clear and understandable product backlog. The product backlog is the backbone of any scrum project. It tells team members exactly what areas of the product they should be working on. For this reason, Product Owners must ensure the product backlog is constantly being updated, furthermore they must prioritize product features based on their value to stakeholders.

Developers are the engine of the Scrum team. They work together to complete product goals. The developers self-administer tasks and are responsible for meeting the goals of each Sprint.

3.1.2 Scrum Events

According to the Official Scrum Guide, “Sprints are the heartbeat of Scrum, where ideas are turned into value” (Schwaber & Sutherland). Sprints are the actual events in which work is done to complete product goals. Each sprint has a consistent fixed length, usually under a month. Because Sprints are done in a consistent time period, it allows for teams to regularly track progress and inspect work being completed. For each Sprint, there is a Sprint Goal that is tied to the Product Backlog and Product Goal. Teams plan to complete this goal within each sprint using four meeting types: Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective.

Sprint Planning is a meeting that takes place before the Sprint and lays the groundwork for what should be accomplished during the actual sprint. While the Scrum Master will direct the meeting, it is meant to be a collaborative time where the Product Owner and Developers work together to create a functional plan for the upcoming Sprint. In this sense, it is important that Product Owners communicate the most important Product Backlog Items and how they relate to the overall Product Goal. During this meeting, the team should address three topics: Why is the Sprint valuable? What can be done in the Sprint? And how will the chosen work get done? It is important to note that Developers should be explicit about their Definition of Done, meaning that they should clearly state their expectations for the completed tasks. Finally, Sprint Planning should take no longer than eight hours maximum for a one-month Sprint.

Daily Scrums are quick fifteen minute meetings held at the same time and place every day. The goal of these meetings is to provide a quick check-in on progress of the work being done within the Sprint. This is important, because if tasks are not being completed on schedule, the team can adjust the Sprint Backlog as necessary to account for upcoming planned work. In Daily Scrums, the team can talk about any accomplishments, impediments, and any other general topics related to the Sprint. Daily Scrums encourage increased communication, which allows for quick adjustments and decisions to be made. In general, this eliminates the need for additional meetings, however Developers can meet throughout the day to adjust their plans for the Sprint.

Sprint Review is an event in which the Scrum Team and stakeholders meet at the end of the Sprint. The Product Owner is typically in charge of inviting key stakeholders to this event. The purpose of this event is for the Scrum Team and stakeholders to have a conversation on what was accomplished during the Sprint, how the environment has changed, and what the next steps to take are. Some topics that can come up include: what went well, what did not go well, progress with the Project Backlog, and reviewing timeline, budget, and product capabilities. It is important to emphasize that this is a collaborative meeting between the Scrum Team and stakeholders, so all members will work together to adjust the Product Backlog and create a cohesive understanding of what should be expected in the next Sprint. This event should be capped at four hours for a one month Sprint.

The final event that caps off a Sprint is the Sprint Retrospective Event. In this event, the Scrum Team reflects on their work throughout the Sprint and looks for ways to increase quality and effectiveness. During this meeting, the team can look at individual performance, interactions, processes, tools used, and the quality of their work. The Scrum Master plays a crucial role in this event by helping team members identify areas for improvement to make the next Sprint more effective and enjoyable. At the end of this meeting, the team should have a plan on what they plan to improve upon and how they will do so in the next Sprint. This event should be capped at three hours for a one month Sprint.

3.1.3 User Stories and Backlog

In the planning stages of a Sprint, the team will work together to craft user stories. User stories are general explanations of a software feature written from the perspective of an end user. The team uses these user stories to provide context and purpose to their work. With user stories, the team knows what they are building, why they are building it, and what value it creates for the project.

Within a user story, a team will create a backlog that will need to be completed for the user story. A backlog is a list of uncompleted tasks for the team to work. Backlogs help the team track progress and show the current amount of work that is needed to be completed. There are two types of backlogs: a Sprint backlog and a product backlog. A Sprint backlog lists user stories that need to be completed within a certain time period. A product backlog lists features that the team would like to implement, but have not yet prioritized.

3.2 Scaled Agile

Scaled Agile Framework is a strategy with specific guidelines on how to implement Agile at an enterprise level. There are five core values that should be emphasized by leadership to effectively use Scaled Agile: Alignment, Built-in Quality, Transparency, Program Execution, and Leadership.

Alignment at every level is an essential piece of Scaled Agile. Through the synchronization of business by the implementation of planning and reflection practices, everyone understands the current state of the business, goals, and how to go about achieving those goals. Furthermore, alignment allows for information to flow both upwards and downwards, which creates a more collaborative and cross functional work environment.

Built-in Quality refers to a company's standard for work. In the Scaled Agile Framework, teams must define what it means for something to be "done". This expectation for quality of work should be woven into all aspects of the business. Within Built-In Quality, there are five key dimensions: flow, architecture and design quality, code quality, system quality, and release quality.

Transparency places emphasis on frequent and manageable planning, so that all members involved with a project are aware of what is going on. Problems that arise can be dealt with quickly and there is real-time visibility on project progress through the backlog.
Program Execution defines the company's ability to deliver quality, working products on a regular basis

Scaled Agile Framework utilizes lean-agile behavior to create better systems for learning, teaching, and working. Lean-agile leadership also entails creating an environment that embraces all core values.

In addition to core values, Scaled Agile Framework uses nine key principles to guide companies in their decision making across functional and organizational boundaries: taking an economic view, apply systems thinking, assume variability, build incrementally with fast and integrated learning cycles, base milestones on objective evaluation of working systems, visualize and limit work in process, apply cadence and synchronize cross domain planning, coach and serve teams rather than command and control, and decentralize decision making (Piikila).

Taking an economic view tells companies to prioritize jobs to provide maximum value, understanding economic trade-offs, and finding ways to increase productivity while reducing costs.

Systems thinking should be applied to understand how tasks relate to the solution, the purpose of enterprise systems, and value streams. By gaining a holistic view, workers can more easily work cross functionally with other teams to optimize productivity.

Assume variability addresses the uncertainty in designing systems and software by using a concept called set-based design. Set-based design is a practice that leaves design options and requirements flexible for as long as possible during the developmental process (Leffingwell, 2021). In this manner, teams can pivot to different options if one is not working. As the team gains more knowledge, they can eliminate options to produce the best possible outcome.

Building incrementally with fast and integrated learning cycles deals with having regularly planned integration points. Through regular and planned integration, teams can look at what works and what does not, which accelerates learning. The goal of this principle is to foster continuous quality improvement and control variability of development.

Teams should base milestones on objective evaluation of working systems. This means that instead of meeting superficial requirements set at the beginning of a project, demonstrating a working system/product provides a more real and accurate reflection of the effectiveness of work accomplished.

Visualizing and limiting work in process helps teams validate that work is benefiting the project in progress. In a software environment, teams can apply this principle by limiting the amount of overlapping work, the complexity of individual tasks, and the amount of work in a given time.

Sprints naturally allow for teams to create cadence with their work. Creating cadence reduces complexity and uncertainty, builds rhythm and routine, enforces quality, and fosters collaboration. By incorporating cross-domain planning, leaders can synchronize cadences between teams to instill consistency across the organization.

Agile implements the idea that teams can only reach their full potential when all members have freedom to use their full skill sets. In this sense, leaders should be servants and coaches rather than bosses.

Teams should have the autonomy to make their own decisions when it comes to the work that they need to get done. Decentralized decision making is effective because it reduces queue lengths and saves upper management time. Instead of micromanaging teams on day to day tasks, leaders should preserve their decision making for important strategic topics.

3.3 Agile at Fidelity

Fidelity uses Agile to manage working across disciplines in groups consisting of members with different expertise. In the Fidelity agile working environment, stories are the basis for tasks that need to be completed. Each story consists of accredited points, which are equivalent to the amount of time/work that it will take to complete. Story points are usually determined by the group, so the value of them can vary among groups. The purpose is to help quantify work to ensure that all members are staying on top of their tasks. Fidelity has been altering their approach to the specific implementation of Agile, specifically they have been drawing influence from the Spotify model of Agile.

The Spotify model, named after the most attractive early adoption company, is not a prescriptive model that insists it is the best way to run a company, but rather was intended to be used as an example of how a company can successfully modify Agile to meet its needs. The major change in this new model is the implementation of Squads, Tribes, Chapters, and Guilds. Squads replace teams as the most basic development unit, containing all positions needed for successful development. While each squad has a particular section of the end product and goals to meet, they are given high levels of mobility and self determination to reach those goals. The point is to allow these subunits to act as best fits them, though they are still held responsible to high authorities (Merryweather, 2022). An example of such a squad within Fidelity was provided by our sponsor. The squad is highly focused on data analysis and consists of ten members. The squad has four each of both data analysts and software engineers, as well as two additional members assigned directly to quality assurance work. This setup provides the squad with the flexibility and resources to work on multiple simultaneous stories while also maintaining quality.

These authorities are the Tribes, collectives of Squads sharing focus on a particular aspect of the project. Tribes exist to ensure that each Squad is progressing smoothly and to allow informal communication between them. This cross-communication is not intended to create dependencies between squads, which would slow progress through bottlenecks and

communications, as each squad is intended to be mostly self-sufficient. The goal is to balance the need and benefits of collaboration with self determinism and ownership, which is where Chapters and Guilds have their place. While Tribes aggregate entire Squads for informal communication, the goals of Chapters and Guilds are to allow more focused areas of cross-communication. Chapters focus on communication between Squad members with similar skill sets, allowing for the sharing of answers to common problems and collaborative innovation. Guilds prompt discussion between workers involved in the same area of a project. Furthermore, guilds are inclusionary as while members directly involved in the area must join the Guild, members who are merely interested in the topic, but hold no direct responsibilities to it, may also join to learn more. This model is not intended to work for every company, and tailoring it to individual needs is a must.

For large corporations like Fidelity, it is important to maintain alignment among groups especially for large projects and initiatives. Agile is designed to create a system where stories roll up into epics, which roll up into sub initiatives, and finally roll up into initiatives. As one goes up in each level the objectives become more general, but collectively form a theme and overarching agenda to help create a well-oiled machine of groups working towards the same goal. Managers and high level executives work with each other to create these initiatives and sub initiatives. For Fidelity, everything is tracked on Jira, a proprietary issue tracking software. On Jira teams post their subtasks and stories, so that all team members may view it. On Jira each Main Story is composed of multiple parts, both internal and external. Internal to the Story are its Background, In Scope, Out of Scope, Requirements.

Each of these sections gives a brief overview of the current goals. Background explains the focus and what needs to be done. In Scope references what is within the control of Fidelity for this operation, while Out of Scope is the opposite. Out of Scope refers to the necessary components of the project that are controlled by other entities, such as what data they provide, and how they provide it. The Requirements section is the largest of these sections. It is a consolidation of the information from the business and what they are looking to do, broken down into actionable items for developers to complete. Externally connected to the story itself are links to prequal for that story, so that employees may both see all that has been done to prepare this story, as well as return to older work for any needs. Jira also provides work logs, history, activity tracking, and commenting space, so that progress on the story can be reported and reviewed. Our

team will be using Jira for the course of this project, formatted in line to Fidelity standards for both work tracking and transparency between our teams, though again, our use of Agile will be modified for our team and project length, in such ways as weekly sprints rather than Fidelity's adoption of two week sprints.

To facilitate their Agile methods, Fidelity teams hold Daily Scrum meetings, typically 15 minutes in length. The Scrum Master is present and facilitating these meetings, prepared to message other teams for assistance and updates, and to help with any access issues. These meetings have each team member give a brief update, stating what they accomplished yesterday and what they will be working on today, as well to report any issues, updates or blockers they experienced. This allows the entire team to be kept up to date on the project, as well as providing a space for more senior developers to take over story points if a time crunch is occurring. Weekly a similar meeting, the Scrum of Scrums, will be held, where the squad leaders will meet and go through a similar process of declaring what has been done, what needs to be done, and reporting any issues. This meeting allows for inter-team communication, to better hear how the project is progressing over all. After the sprint is completed a more in depth retrospective will be completed, though as Fidelity uses two week sprints these are not weekly as in other companies.

The last traditional Agile Scrum meeting used by Fidelity is the Sprint Review, though its implementation will depend on the team. Our sponsor's team for example, rather than do traditional meetings with the project stakeholders discussing what was accomplished and what will be done next, performs Scaled System Demos for the stakeholders, often with up to a month between these demos to the higher-ups. Next Sprint Planning Meetings are held by squad leaders, where they review backlogs through Backlog Refinement, determine story importance and prioritization, and distribute them to teams with capacity. Our team will follow a similar approach, with Daily Scrums, Scaled System Demos and both Sprint Reviews and Retrospectives, though we are unlikely to perform Scrum of Scrum as we are a single team.

3.4 Data and Computer Science Techniques

3.4.1 Anomaly Detection/Machine Learning

Anomalies occur in the data where the beginning day assets do not equal the ending day assets after it has gone through the data flow. The team explored different algorithms to detect an

anomaly, including STL decomposition, Classification and Regression trees, and Detection through Forecasting.

To completely solve the problem with anomalies, it is not enough to simply detect an anomaly. It would be ideal to find the cause of the anomaly, fix the anomaly, and then take preemptive measures to reduce the number of anomalies that happen in the future. Machine learning is a big part of the future steps of the project. Finding the cause of an anomaly will allow analysts to take specific actions to reduce the number of anomalies that happen in the future. In order to find out why certain anomalies happen in data, the team proposes to look at the anomalies in consecutive months and examine the data attributes. The goal is to find a correlation in the data flow and to see if certain attributes are associated with anomalies.

3.4.2 Data Cleaning/Patching Data

During the course of this project, we had access to Fidelity regulated fund's assets and transactions. This data was not exportable to Excel, therefore most of the calculations were done using SQL on the Snowflake platform. Some of the simple calculations done were finding the percent and dollar difference between the starting and ending balances. Another error preventive measure was, when calculating the percent difference for every count, we had to make sure that the dollar difference was not zero, or else we would get a dividing by zero error for our calculation.

At times, the team joined multiple tables together in order to compare ending and beginning balances for customer transactions. This required us to complete self joins in SQL based on a certain criteria. Because we wanted to compare the previous day balance to the current day balance, we joined the table with itself where the date of one table was equal to the date of another table plus one. This ensured that we had the previous date and current date in the same row. To do so, we had to cast the date into a date field, as the column was recorded as an integer. We utilized substrings and the to date function to change the date field.

To clean the data, we checked if every column was missing values. If more than 15% of its features were missing we would remove the specific instance. Unknown entries were marked as 0 if no data instance was found after filtering.

3.4.3 Feature Engineering/Feature Importance

Since the assets and flows data consists of a wide and diverse set of continuous and discrete features, it is important to properly extract the most meaningful features for a downstream anomaly detection. Doing so will guide future experiments as to what features are the most valuable for up-sampling with more synthetic instances. In order to find these important features, we used both manually selecting features and random forest (RF) feature importance (Gunn & Rogers, 2006).

Random forest is a method that uses an ensemble of decision trees generated from a randomly divided dataset. A set of such classifier trees forms a forest. Each individual decision tree is generated using feature selection metrics such as information gain criterion, gain ratio, and Gini index for each feature. The algorithm first creates random samples from the given data set. For each sample, it builds a decision tree and gets the prediction result using this tree. After gathering predictions, the algorithm gathers the votes from every prediction tree and selects the prediction with the most votes as the final result.

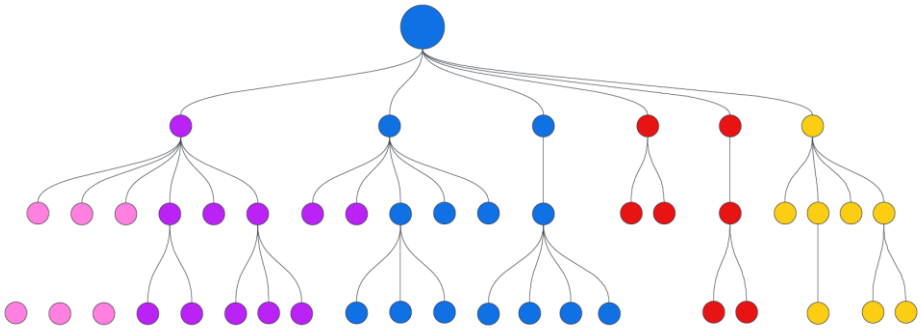


Figure 2. Random Forest Representation

Random forest offers a good feature selection criterion. Scikit-learn provides an additional variable that shows the relative importance when using a random forest model. Relative performance shows the contribution of each indicator to the prediction. The library automatically calculates the relevance score of each feature at the training stage. The resulting value is then normalized so that the sum of all scores is one (Pedregosa et al., 2011).

Random Forest is considered to be a highly accurate and reliable method because many decision trees are involved in the prediction process. It is less likely to be impacted from overfitting. The main reason is that Random Forest uses the average of all predictions, which

eliminates bias. RF also calculates the relative importance of features, which helps in selecting the most significant features for the classifier.

When utilized in the context of RF, the feature importance algorithm first quantifies how much each feature contributes to the final prediction of a decision tree, and then determines the average values of these contributions across the forest. More specifically, the RF feature importance is computed by measuring the degree to which each feature reduces the Gini Index,

$$Gini(\omega) = \sum_{k=1}^{K} \omega_k(1-\omega_k) = 1 - \sum_{k=1}^{K} {\omega_k}^2 ,$$

where k-total number of features considered, and ${\omega_k}^2$ represents sample weights. Moreover, within a single tree's internal node m, the feature importance γ of x is

$$\gamma_{jm}^{(Gini)} = GI_m - GI_l - GI_r$$

where GI_l and GI_r are the Gini Index of the two child nodes after a split respectively. Given that a feature x appears in a decision tree i in nodes M, γ of x in the i-th tree is defined

$$\gamma_{ij}^{(Gini)} = \sum_{m \in M} \gamma_{jm}^{(Gini)}$$

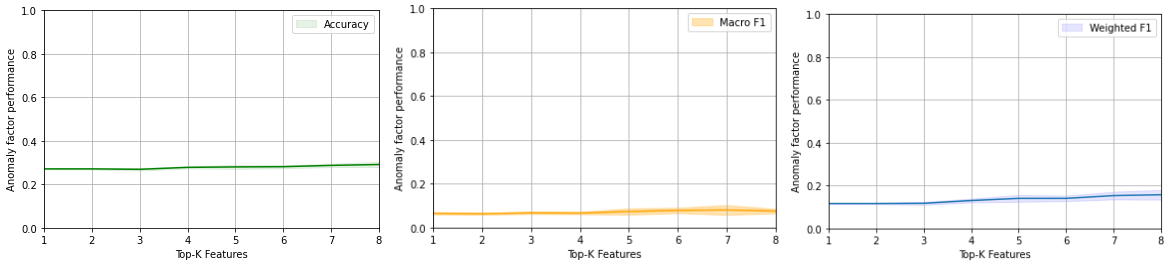
Finally, we normalize the values of γ by dividing a feature's importance by the total sum of all feature importance values.

$$\gamma_j = \frac{\gamma_j}{\sum_{i=1}^{C} \gamma_i}$$

From there, we perform stepwise feature selection to determine how many of the top−k most important features to consider for our final baseline feature input for a hypothetical downstream anomaly detection model.

To find the most important features, we ran a set of stratified random forest feature importance tests for each of the eight factors. The random forest indicated that the three most important features were: 'NON_CUSTOMER_NET_FLOWS', 'OTHER_NON_CUSTOMER_CHANGE_IN_ASSETS', 'CUSTOMER_NET_FLOWS'.

By finding the minimum number of features necessary until reaching asymptotic performance in classification when using real data, we would know what features are most important for up-sampling with synthetic generation as well as input for anomaly classifiers in our experimental study.



(a) *(b)* *(c)*

Figure 3. Plots a-c. From left to right: accuracy, macro-f1, and weighted-f1. Regression model performance when using stepwise selection of features in Assets-flow bridge table.

These results corroborate with our sponsor's belief that TA data was where anomalies would be coming from. We started working with the TA/FA asset bridge table, however our results resulted in our next steps: looking for anomalies in TA data.

3.4.4 Time Series

Fidelity database consists of time series data, which is data that changes over time. Time series analysis is based on the analysis of data in which there is a time dependence in order to detect statistical dependencies and other characteristics. Time series can be used to make predictions based on using a trained model to predict values in the future based on observed values in the past (Computer Science Center, 2018).

When working with time series data, we decided to split it into smaller segments of different sizes. We started by performing simple regression models on the data from one day, two days and a week.

Segment regression is a regression analysis method in which the independent variable is broken into intervals, and each interval corresponds to a separate line segment. It does this by breaking the data set into segments, each with a specific range. It then completes a linear regression on each of the segments. Segment regression is useful when explanatory variables grouped into different groups show different relationships between variables in those regions. The boundaries between segments are breakpoints (Orac, 2019).

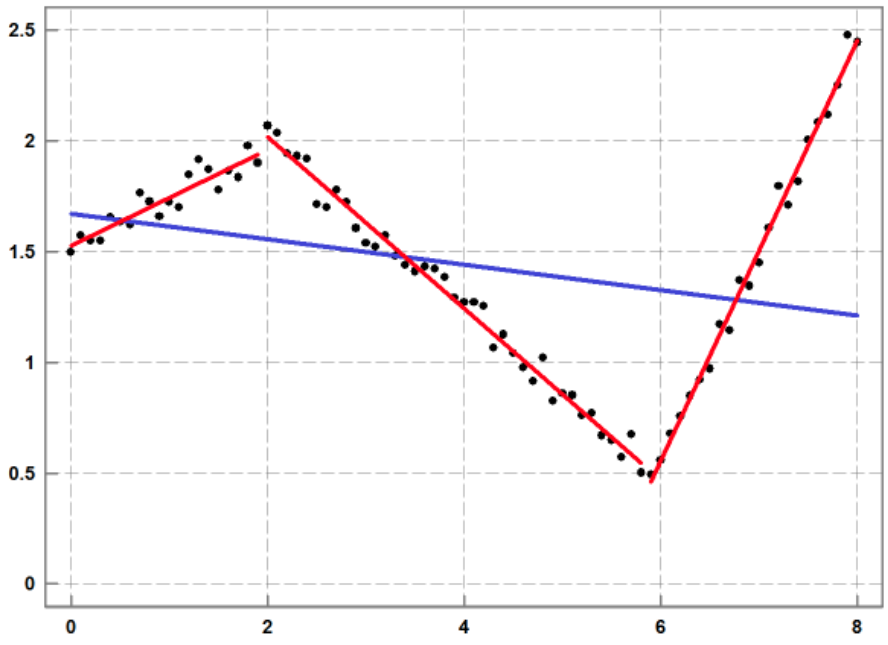


Figure 4. Segment Regression and Linear Regression

This graph demonstrates how we are able to get the linear regression of each section. The segmented regression is shown in red and the linear regression is shown in blue. It is not practical to run a linear regression on a sample like this. A segmented regression will give a much more accurate answer by breaking the large dataset into smaller pieces. This method will also help find differences in the data. For example, sharp changes in the data will result in a different segment. The algorithm for this method grants the required accuracy for the model, using the smallest number of segments possible.

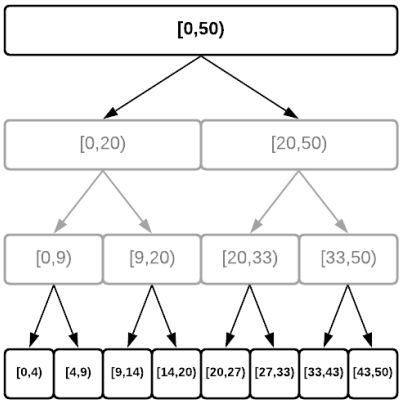


Figure 5. Splitting the Dataset

This figure shows the idea of splitting the dataset into segments. The top ([0, 50]) shows the entirety of the dataset and as we move down it, the diagram is splitting it into subsections. It then calculates the linear regression on the subsection. With each split, the model is getting more accurate (Stadnik, 2022).

4. Software Development Environment

4.1 Background/Access Software

The team used Citrix, a remote desktop application, to access the Fidelity network while working remotely from the WPI campus. All team members were given Fidelity credentials to accompany Citrix. All data processing and intra-Fidelity communication was done through remote desktop connection to the Fidelity network.

4.2 Agile Software

The project management software that we used was Trello. Trello is a great tool to keep track of the progress of the current sprint. In Fidelity, sprints last two weeks, where the senior developers also have a say of what the final story point should be for each task. We had a Trello board for every sprint, and our sprints lasted one week, making a total of seven sprints. The sections in our Trello board included: To Do, Doing, Needs Review, Done, Questions, Helpful Links. These sections helped keep the team organized as we knew what each group member was working on and we can assign work. Keeping in line with Agile, we also assigned story points at the beginning of each sprint, and then logged all progress with a sprint retrospective at the end of the week.

At the end of the second week Sprint, the team received access to Jira on the Fidelity network. Jira is an issue and project tracking software, which allows for the centralization and easy sharing of information regarding tasks, blockers, and delegation. The team began using Jira to organize our sprints, as within Jira weekly sprint boards would allow for the organization and assigning of stories, as well as a rolling product backlog which allows for review of previous tasks. We quickly found that JIRA is more effective for collaboration between squads. The team decided to stick with Trello for the project, because of its ease of use and simplicity as compared to JIRA. With JIRA, the team also ran into access issues and did not find much use in the additional features offered.

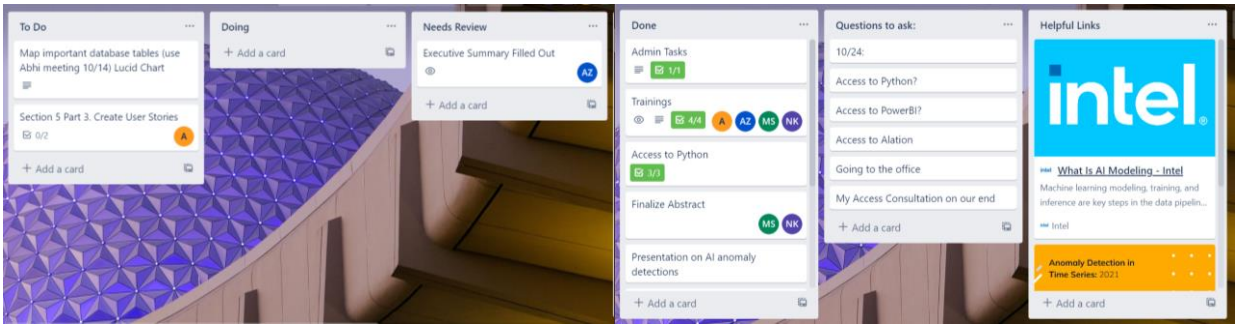


Figure 6. Example Trello Board

4.3 Data Access and Analysis

The team had access to various software platforms, such as Snowflake, PowerBI and Python. We viewed the assets and flows dataset through Snowflake. Snowflake is a cloud computing program that consists of three main functions, Database Storage, Query Processing, and Cloud Services. Snowflake takes care of how the data is stored as well as the authentication and query processing from the services provided by the cloud. In Snowflake, we used SQL to query data.

PowerBI helped us visualize and manipulate fund and transaction data pulled from SnowFlake. We applied the filter tool to this large data set to help build another asset bridge for the TA data.

For sharing code within the team, we used a private repository on Github. This allowed for a secure sharing of the algorithms. Since no Fidelity database was uploaded in Github, this platform complied with the privacy and security agreements. The main code language used was Python 3.10, which was preferred as many machine learning models' libraries are built in Python, such as pytorch, scikit-learn, and tensorflow.

Excel was used for initial data exploration and visualization, after the data had been pulled from Snowflake. We exported the data and used in-built Excel tools, such as functions and pivot tables, to manipulate the data. Through Excel, we were able to replicate asset bridges between Beginning and Ending day assets on the Customer Portfolio View (CPV) level, with the purpose of detecting when anomalies occurred. No additional Excel Add-ons were used in our analysis. Later and larger scale data analysis was done within Python, once the team was more familiar with the structure of the data.

5. Software Requirements

5.1 Requirements Gathering

After setting the project goals with the project sponsor in the first week of the PQP, we came up with the necessary tasks for the successful project completion. The data analytics team managers and software engineers helped guide the team in creating the project requirements.

In the initial weeks of the project, the team met regularly with two data experts. In these meetings, they walked us through the Snowflake and PowerBI databases. Specifically, they made suggestions on possible tables to investigate and example activities that the team could do to gain an understanding of the data. Through these walkthroughs, the team explored the data issues first hand, while piecing together the requirements that would be needed and possible solutions for the project. Our project liaison helped with any urgent issues and connected us to the Fidelity data and software team. The team connected with our project sponsor biweekly to update him on our progress and ensure that the work being completed was in line with project goals.

5.2 Functional Requirements

Our team created a software solution to create a look through table by patching Mutual Fund (FA) and Transfer Agent (TA) models. Using a table, the team charted out the functional and nonfunctional requirements for the software:

Functional Requirements	**Non-functional Requirements**
Have the ability to merge FA and TA models into one look through table that detects the discrepancies in the accounts	Keep the layout of the manually created look-through table
Have the ability to export the look through table into a .csv or .xlsx file in PowerBI	Reduce labor put on Data Analytics team to give space for higher value activities
Keep the tool’s batch processing ability	
No data loss appearance when patching the data	

The second assignment was to fix the software anomaly that resulted in missing data in the look through table. The table has a valuable data set that the company presents in the annual report as well as predicting future trends. The requirements for anomaly detection machine learning network were:

Functional Requirements	**Non-functional Requirements**
Evaluation performance shows improvement in the anomaly detection	Our second project did not have any non-functional requirements.

5.3 Epics and User Stories

Our team created a set of user stories before creating stories in Trello. The stories were created by talking to several teams that interact with TA and FA models as well as individuals who manually patch the look-through table. Our epics were:

Epic 1: Understanding the Fidelity Data Structure	
As a user:	I want to be able to easily find the data I am looking for so that I can use it quickly.
As a developer:	I want to be able to pull data from the right location so that I do not have to consider bad data
User Stories:	I. Gain access to Snowflake II. Learn to query in Snowflake III. Perform transformations and calculations on data within Snowflake IV. Pull Snowflake data into Excel and Python and operate on it there.

Epic 2: Recreate TA/FA table provided to us	
As a user:	I want to be able to see where the money in the fund is flowing from/to so that I understand the condition a fund is in.

As a developer:	I. I want to identify the severity of anomalies in the bridge so that I can prioritize patching the most harmful. II. I want to be able to identify where the bridge is breaking down so that I can patch the bridge in that spot as quickly as possible.
User Stories:	I. Pull correct data II. Create data table in excel with appropriate calculations III. Detect anomalies in data (at the transaction and fund level) IV. Preform anomaly detection in Excel

Epic 3: Create Bridge Process for Pure TA Data	
As a user:	I want to see the total beginning and ending assets for the TA data to detect anomalies
As a developer:	I want to build a bridge between the TA data to spot errors quickly
User Stories:	I. Map the Table via LucidChart (Entity Relationship Diagram)

Epic 4: Anomaly Detection on TA Data Bridge	
As a user:	I. I want to easily detect what the anomaly is in the data so that it may be patched as quickly as possible. II. I want to identify what caused the anomaly to have the anomaly patched properly and prevent the same issue in the future
As a developer:	Create a model for analyst to go to for the TA data, that will show and predict anomalies in large data sets
User Stories:	Not started yet/Future work

6. Business and Project Risk Management

6.1 Business Risk

As Fidelity is a private company, they are able to take different risks than a public one. Public companies are bound by fiduciary responsibilities to their shareholders, creating the possibility of neglecting future sustainability for short term wealth generation. Fidelity, as a private company, has more freedom in its actions. Fidelity's goals are determined by its board and CEO, allowing them to make investments in industries even when it is against the collective drive of the market.

6.1.1 Operational Risk

Dealing with regular data errors leads to longer periods of reconciliation and the knowledge of prevalent errors reduces consumer confidence in Fidelity. Fidelity is currently employing a team of six to eight associates to deal with these errors, but due the size of Fidelity managed assets, they must focus on the most egregious anomalies. In addition, many other business units rely on the assets and flows data, so if the publication of the final data set is delayed, then work across the company will be delayed, contributing towards a large operational risk.

Through the use of system integration, development, UAT, and production environments, Fidelity tries to organize work that is being done on the databases. In system integration and development environments, developers can make changes to the databases without affecting any important data flows and disrupting any production data. In the UAT environment, developers can test changes made and make sure that the data is ready for production. The production environment contains finalized data that is ready for downstream analysis and business operations. Production risk arises when, despite stringent testing, there are still errors in the production data.

Relating to errors in the data, there is operational risk in the misidentification of error origins. Our project liaison gave an example about an incident where his team identified errors within the dataset. The team believed that the error was a result of their processes, and therefore went about looking for the source of the error in their code. The error was actually a result of the

misprocessing of data by an accounting team. The lack of communication between squad leads cost our liaison's team time and effort in searching for a nonexistent error in their code. He suggested better communication between squad leads to mitigate these types of oversights. Because data flows in streams from squad to squad, it is essential that any errors are quickly identified, especially upstream, to minimize the operational risks further downstream.

Currently, Fidelity is in the process of shifting all of its data silos over to Snowflake. Before Snowflake, the company did not have a standardized data collection and storage infrastructure. This made it hard for analysts to derive any insights as the data was so segmented. Furthermore, having many types of data platforms unnecessary complexity and room for error when trying to create aggregated data. By shifting all data silos over to Snowflake, Fidelity is reducing the production risks associated with unstandardized data storage. Furthermore, Snowflake will allow analysts to get more value out of the data to create better insights to drive business growth.

6.1.2 Financial Risk

The process of identifying errors and delaying production time due to errors costs Fidelity associates time and resources. In this sense, the more errors in the data, the higher the cost it is to maintain the data. Furthermore, the more downstream an error gets unnoticed, the more difficult and costly it is to fix. By reducing the amount of errors in the data, Fidelity can minimize this piece of financial risk. If an error goes completely unnoticed, then the audit team will have a very tough time piecing together accurate financial statements. This emphasizes the importance of finding the anomalies early and identifying the cause, in order to keep financial risk to a minimum.

6.1.3 Reputational Risk

There is a reputational risk associated with having errors in the production data. If these errors appear in customer facing products and accounts, then they can seriously damage Fidelity's reputation. Customers want to know that their data is safe and accurate. When Fidelity cannot guarantee this, then customers will lose trust in the company and think twice about investing with Fidelity. For workers, if they are constantly dealing with bad processes that make their work difficult, then they might lose confidence in the company as well.

6.1.4 Information Security Risk

Dealing with confidential financial information requires rigid data security. In our project, Fidelity took many steps to mitigate any information security risks. Firstly, they recruited the team from WPI, a highly accredited institution with trusted former employees who could act as contacts in the event of damage. Fidelity had background-checks conducted on all team members and mandated information security training. The company also required the use of a secure remote desktop connection to access the Fidelity network.

At an associate level, Fidelity implements training and guidelines to protect confidential information. The company requires all associates to undergo information security training. In these trainings, they teach associates what confidential data is, proper uses for confidential data, and the appropriate situations for sharing that data. The company also has a set of guidelines that employees must agree to following. For example, they place restrictions on sending and receiving files to/from external parties. When confidential information must be sent over email, employees must use built-in encryption tools to secure the email. There is security risk involved if associates do not follow the guidelines for dealing with confidential information. The enterprise security team monitors and ensures that the guidelines are being followed to minimize this risk.

When working with large datasets, there are levels of access that need to be requested. Associates should only have access to data that they need for the project that they are working on. For example, in Snowflake there was a large set of warehouses that we could choose from, however our initial access only granted us permission to two of them. When we realized that we needed access for another warehouse, we needed to request permission again. In the request, we needed to state our reason for usage and have the request approved by our manager (project liaison). Fidelity wants to limit the access to data because with increased exposure comes more risk for data breaches. These levels of access are another way that Fidelity manages data security.

6.2 Project Risks

Throughout the course of the project, the team dealt with various project risks.

6.2.1 Resource Risk

Fidelity and their selected sponsors gave the team freedom to be creative with a solution to their business problem. While the hands off approach allowed the team to take full initiative of the project, it also proved challenging. In many instances, especially at the beginning of the project, the team felt that little progress was being made and struggled to find the perfect direction to focus progress, particularly in regards to fully grasping the context, scope, and definition of the business problem. Without the correct business context, we were creating a potential resource risk, as more time would have to be spent correcting the course of the project. To manage this risk the team ensured that we had direct communication with our sponsor and made use of any chance we were given to present our progress and ask for feedback and direction.

6.2.2 Technical Risk

Technical risk was a prevalent topic throughout the duration of the project. Originally, the team thought about producing a new application to report on TAFA anomalies. We quickly realized that bringing in new technologies would pose a greater risk than producing a final product in an already used application like PowerBI. The first reason contributing to risk would be the learning curve that it takes to get used to the new application. Associates want to work with applications that they are already familiar with. Secondly, we learned that there would be implications with integrating a pipeline for the new application and the Snowflake data. It makes more sense to use an already established data pipeline. Third, because this application is very specific to one business problem, it may not be worth spending time to update it in the case that the application becomes outdated. For these reasons, we concluded that unless absolutely necessary, the team should stick with already existing applications to reduce technical risk.

Because the team would be attempting to apply new technologies to Fidelity's systems, there are risks to the overall system, especially in regards to quality assurance testing. As any product was completed in such a short time frame, there was near certainty that edge-cases were missed and that there were blindspots that the team had not been exposed to, limiting the effectiveness of the model in unknown ways.

6.2.3 Business Knowledge Risk

As much of the project involved identifying areas of potential risk and the planning of development for systems to manage the anomalies in the data, there exists the risk that this knowledge and planning could be lost and future developers will be tasked with rediscovering this information. To manage this knowledge continuity risk, the team has endeavored to be as transparent as possible with our sponsors regarding what we worked on and what was produced, as well as future implementation ideas of our work. We tracked and recorded all research and processes in this paper, with more notes stored in a shared secure folder.

From a knowledge acquisition risk standpoint, the team had to think about how the new insights would improve current processes for business analysts and how easy it should be for them to access the information. If the team created a program that gave false or inaccurate information, then the project would hurt the business more than it helped. Furthermore, the team wanted to create valuable insights on the data. If the program generated data, but the data had no purpose, then the project work would have been done in vain. To manage this risk, the team made sure to clearly communicate the program potential and discussed how the data produced would be used to help analysts.

7. Design

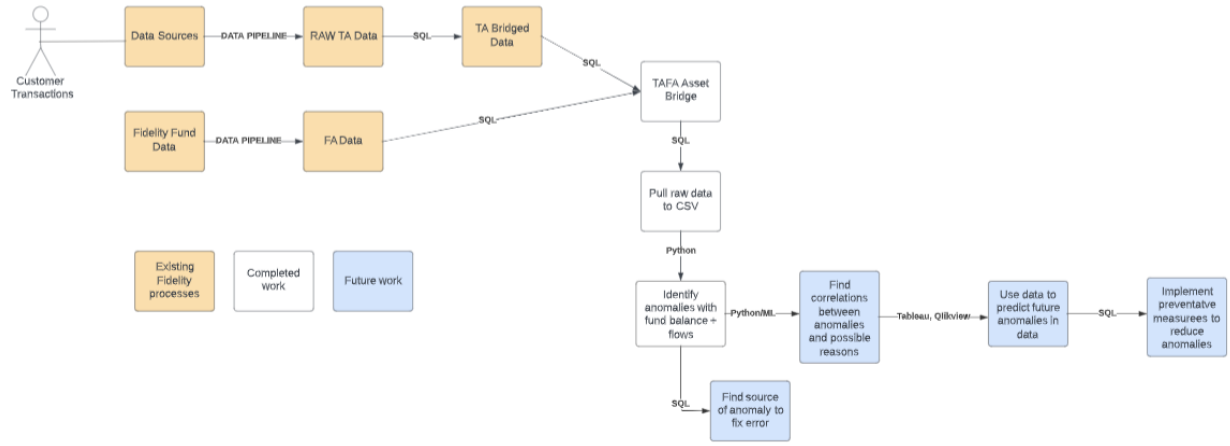


Figure 7: Project Diagram

In the figure above, the existing Fidelity processes are shown in orange, our completed work shown in white, and future work shown in blue. The raw TA data table is in Snowflake and consists of transactions that customers have made. There are many data sources that feed into the Raw TA Data, whereas only the Fidelity Fund data feeds into the FA Data. In Snowflake, there is a TA Bridge Data Table and combined with the FA Data, creates a TA/FA Asset Bridge. We extracted raw data into a CSV file for the python program. From here, we used python to detect anomalies in the fund balances based on starting and ending amounts. Our future work includes using python and machine learning to find possible correlations between anomalies and the data flow. More importantly, we would want to find the source of the anomaly and thus fix the error. The final steps would include using the data to predict when anomalies will happen in the future and implement specific tactics to minimize anomalies.

7.1 Asset Bridge

The first product that we worked on to show the company was an anomaly detection program for the TA/FA asset bridge. The goal of this product was to determine whether the beginning day assets matched the ending day assets after all data flows were calculated. These flows take the form of individual transactions, market action, dividend payments, among other changes that happen to accounts on a day-to-day basis. Each fund account is recorded in a table

with fourteen columns. Each column contains information regarding the transaction, such as identifying information or associated funds at different levels. Six columns contain flow data.

To find anomalies in the TA/FA asset bridge, we queried Snowflake to pull all transaction data from a specific date and export it to Excel. In Excel, our first task was to explore the data and manually search for anomalies in the data. To verify this, we began with verifying that the sum of the flows would equal the change between beginning and ending assets. If the sum of flows and the beginning assets equaled the ending assets, then that fund was marked as good. Otherwise, we looked into the deviance from expected values. We first calculated the dollar difference between the sum of flows and the beginning assets and the recorded ending assets of that day. The dollar difference was then used to provide us with a percent deviance of that anomaly, and together the dollar difference and percent difference would allow us to categorize and rank the significance of each observed anomaly.

For a deviation to be marked as significant, it must breach two separate thresholds, an absolute threshold and a relative one. Due to the volume of sales within Fidelity, small anomalies, on the scale of only a few thousands of dollars for example, are not worth the opportunity cost of ignoring larger anomalies, such as those in the millions or billions. Second, the relative scale of the account must be considered. Although a large amount of money may be missing from an account, a few million for example, if that account regularly trades in the multiple billions, a loss of less than .1% is acceptable and even expected, where as the same amount lost in an account that does not expand past the hundred million mark would be a much larger call for concern. These anomalies were ranked in terms of their significance for the company, taking both measures into consideration.

Figure 8. Asset Bridge

Above shows a simple figure representing the asset bridge. When this equation did not hold true, then we flagged the portfolio number associated with this and then ranked the anomaly based on its severity.

To support analysts with their anomaly detection in the asset bridge, our team proposed the idea of building a single platform that all analysts can use to receive information regarding the existence and severity of anomalies. This platform would be capable of receiving a day's account and transaction data, use that data to detect anomalies in accounts, and then be able to offer suggestions about the possible origin of the anomaly.

As part of the Python program, we first started by making three new columns in the asset table. The first column was called "Expected Ending" where we calculated what the actual ending balance should be. From there, the next column was called "Dollar Difference", which represented how far off the actual ending was from the expected ending, in dollars. We repeated the same steps in a new column called "Percent Difference". We then set two thresholds in our program, dollar and percent threshold. Here, we added another two columns that returned 'Yes' or 'No' depending on if the anomaly reached the certain thresholds. A new table was returned with only the anomalies that reach both thresholds. The next step was to add a drill down feature that tells us what specific transactions had an anomaly. From there, we gave suggestions on how to fix the anomaly based on a new column called factor. The factor column told us what aspect in the data flow caused the anomaly. See Appendix A to view this Python code in more detail.

7.2 Entity Relationship Diagram

Using the TAFA asset bridge table code, the team created an ERD to help map out the data sources used to create the table. Because many of the tables used to create the asset bridge table consisted of twenty plus columns, the team decided to use only the columns that were used in the code. For this reason, many of the tables do not have primary keys, as those keys were left out of the code. In addition, the code uses conditional statements to filter the data and then mathematical operations to create columns in the asset bridge table. In this sense, the columns of the asset bridge table are pieces of the other tables stitched together. Although this activity was not directly related to anomaly detection, it helped the group understand the data flows into the TAFA asset bridge table, which helped with knowing where data errors could be occurring and whether it was a data or coding issue.

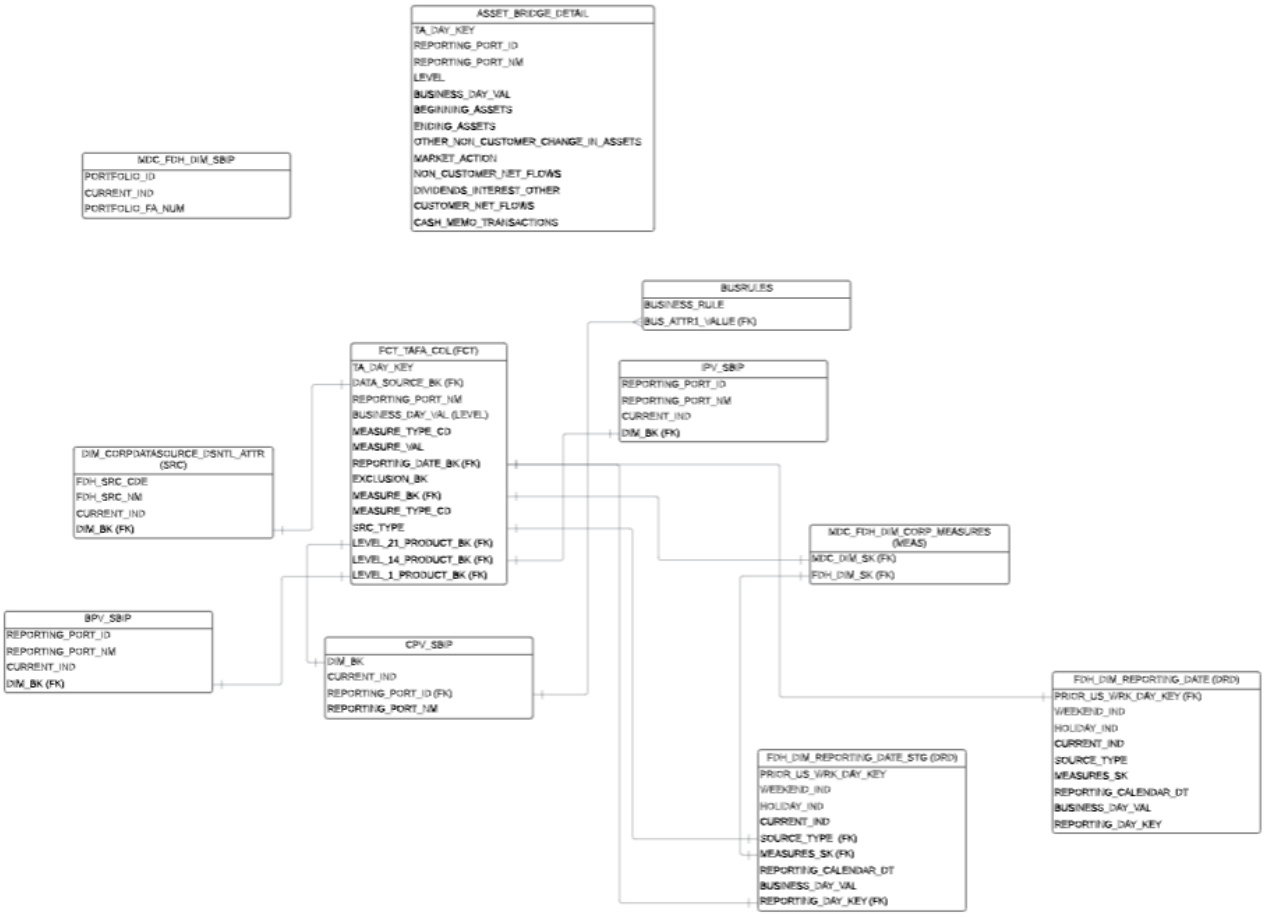


Figure 9. ERD Diagram of TA-FA Table

8. Software Development

8.1 Scrum Details

Before the team started official work on the project, we laid out guidelines for our Agile working methodology. Austin served as the scrum master. Mitchell played the role of product owner. Nathan and Aru were the group's developers. We booked a regular in person meeting time from 10am to 1pm, Monday-Friday. On Monday, we would begin the working session with sprint planning on Trello. Tuesday-Thursday, we hosted a daily scrum to recap any work completed the night before. On Friday, we would host a sprint recap in the morning to reflect on the work completed during the week. We took notes to summarize what was completed and jotted down any significant improvements that were made or could be made in the future.

8.2 Sprint 0

8.2.1 Summary

Week zero of our project started on Monday, October 24th. On the first day, our team met to discuss the plans for the week as we assigned tasks on Trello. Our team met for daily multi-hour sessions, to help each other complete tasks, which mostly consisted of understanding and working through the data on Snowflake. This week consisted of many meetings and daily communication with our sponsor and other Fidelity personnel. The goals of this week were to run sample queries on the data, understand the asset bridge that we are building, or simply make sure we have access to software. As a lot of this week consisted of understanding the data and administration tasks, we considered this week sprint zero. Our team still met daily and updated the paper when needed. A list of tasks that were completed and not completed are shown below.

Story	Subtasks	Points (1-5)	Assignee(s)	Completed?	Date Completed
Fidelity	SH Training	2	Everyone	Yes	Oct. 28

Trainings	-Data Security Training -Fidelity Compliance				
Research AI Anomaly Detection	-STL Decomposition -Classification and -Regression Decision Trees (CART) -Detection Through Forecasting	4	Aruzhan Nathan Mitchell	Yes	Oct. 25
Week 1 of Paper	- Introduction - Executive Summary - Background - Abstract -Week 0 retrospective	2	Everyone	Yes	Oct. 28
Admin Tasks	- Set up Trello - Ensure Fidelity Access - Finalize Schedules	1	Everyone	Yes	Oct. 25

Table 1. Sprint 0 Story Points

8.2.2 Retrospective

Overall, this week proved to be productive for the group as seen in the tasks completed. The morning meeting times worked well for the group. We scheduled the meeting times at either 10 am or 11 am for typically two hours. In those working hours, the group discussed Sprint tasks, worked collaboratively on tasks, and met with the project sponsors. Any work left after the two working hours (usually individual work) was divided by the group members to complete at their convenience for the next day's meeting.

The scheduled meetings with our Professors helped most with identifying which questions to ask our sponsors to resolve our issues. Their high level understanding of business

and data science subjects helped guide us in the best path for our work. In this sense, they saved us time from going in circles by providing a sense of direction.

There was one meeting with our sponsors that two group members were unable to make, the meeting also did not get recorded. Thus, the two group members that did attend needed to recap and fill the other group members in and may have missed details along the way. In the future, we will be more cognisant of scheduling around conflicts, as well as ensuring that all meetings get recorded.

8.3 Sprint 1

8.3.1 Summary

This week, from October 31st to November 4th, marked our first official sprint. This week, we prepared our first product to show Fidelity. This product was the asset bridge where we attempted to reproduce the results that were given to us from a principal analyst. To check the accuracy of our model as well as answer any remaining question we had about the data flow, we set up a meeting with our technical sponsor on Wednesday. During this meeting, our sponsors informed us that they believed we had the right idea for the asset bridge, but will schedule a meeting with the data analyst on Monday November 7th, to get his thoughts. During this meeting, we also talked about the potential of meeting with another analyst, who we previously met to discuss what adjustments would have to be made based on the anomalies that we found. We would then rank the severity of the anomalies based upon how much of an adjustment would have to be made to fix them. Thursday, our team got access to Jira and were given a full tutorial. Our team plans to implement Jira the following week for Sprint 2. Finally, on Friday, our group worked together on building a PowerBI model and discussing moving the Excel asset bridge to Python.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Mimic the Assets and	- Create a query - Explore the data in Excel	4	All	Yes	Nov. 4

Flows Table presented by the Data Analyst	- Mimic Variance tabulations				
Attempting ARIMA on the various data points	Take anomaly data from different points and perform the ARIMA analysis to see the pattern	5	All	No	No longer priority consideration
Paper	-Review Professors' comments -Expand previous sections with new content	2	All	Yes	Nov. 4

Table 2. Sprint 1 Story Points

8.3.2 Retrospective

Overall, our team felt like we had a productive week. This is mainly due to the fact that we were able to show our sponsors and advisors the asset bridge that we have been working on. This is the first product that we were able to show and thus we were able to receive feedback and gather the next steps. Another positive takeaway from this week is that during the middle of the week, when we felt like our team needed help, we immediately asked for a meeting with our sponsors and included a detailed write up about what we had trouble with. This allowed us to set up a meeting the following day, which continued our progress. The weeks move fast so we did not want to get held up by a blocker that can be fixed in one meeting. Our team met every day this week as well, mostly in person, which continues to be the best way to make constant progress.

Last week one of our largest failures was a lack of documentation and communication. This manifested it not having easy access to recordings of meetings or reference examples that would clarify issues and help alleviate blockers. To remedy this, we have changed our team

policy to improve our level of documentation and ensured that all meetings we attended were recorded and uploaded in our shared workspace.

As long as we continue to meet everyday and ask for help when needed, we will be on the right track.

8.4 Sprint 2

8.4.1 Summary

Monday marked our bi-weekly meeting with the analysts at the company to show our progress. The Fidelity Head-Sponsor was unable to attend the meeting, but we were able to meet with the data analyst. We were able to show him our progress in Excel and as well as pitch the idea of a platform for dealing with anomalies in Assets and Flows. Based on what they told us, we started making the adjustments to the asset bridge in excel. On Wednesday, we had a meeting with an analyst, to discuss the query to build the asset bridge. The same day, the group completed the asset bridge using python that was then shown to our sponsor on Thursday. Overall, our sponsor was interested in our demo and set up a meeting for the following week to show more analysts. We ended our week by starting to map the tables involved in Snowflake by building a diagram in LucidChart.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Clean the query	- Fix a query - Explore the data in Excel - Mimic Variance tabulations to consider dollar value and variance percentage	4	All	Yes	Nov. 11
Create a map	- Build a diagram that shows how Assets Table	3	All	In Progress, waiting for	

	was built			clarifications from Prof. Wong	
Process the Assets and Flow query	- Reach out to Azar to receive a query - Process and understand the query	3	All	Yes	Nov. 11
Collect test and train data	- Pull train and test data from Snowflake from for 20221014 to 20221030	2	Nathan, Aruzhan	Yes	Nov. 11
Neural network training	- STL decomposition training - Focus on Fidelity funds	5	Nathan, Aruzhan	Yes	Nov. 4th
Paper	-Review Professors' comments -Expand previous sections with new content	2	All	Yes	Nov. 11

Table 3. Sprint 2 Story Points

8.4.2 Retrospective

During Week 2 we were able to present our idea to the data analyst and start researching on python web application and ARIMA neural network. After meeting with Azar we received a query that was used to build the Assets and Flows diagram, which allowed us to start creating a facade diagram-map. This week was productive in that we continued to chip away at understanding the data structure for the asset bridge. Creating the Assets and Flows diagram will help us be able to understand data flows, which will help in creating processes for identifying errors that are fed into the Asset Bridge table. Overall, we did a good job of communicating our

work and ideas to the project sponsors and asking any questions to further set us on the right course.

8.5 Sprint 3

8.5.1 Summary

On Monday and Tuesday of this week, we worked on our python program and presentation that we were going to have to give on Wednesday. Wednesday, the team went into the Boston office to meet with our sponsor and present the progress over the start of the term to a group of analysts. The group felt the overall outcome of the meeting was positive and it gave us good direction for the coming weeks. The analyst said that we had the right idea, but were using it on the wrong table in Snowflake. They believe that it would be beneficial to complete a similar demonstration, but on a share level. On Thursday morning, our Sponsor emailed some associates that the Head-Sponsor mentioned in the meeting to get the proper name of the table that we should be working on, as well as a Snowflake to Python pipeline. While waiting from a response to get the proper table name, we made updates to the paper, as well as our ERD diagram.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Add predictive and identification columns to python program	Add likely source of error column Add drill down feature based on CPV Add column for account size labels	7	Nate, Aruzhan, Mitchell	Yes	Nov. 15
Demo program to Head-Sponsor	Create presentation Go over presentation with Professors	4	Nate, Aruzhan, Mitchell, Austin	Yes	Nov. 16

Finish Diagram Map	Add primary and foreign keys to map	3	Nate, Mitchell, Aruzhan	Yes	Nov. 17
Weekly Paper	Continue work on paper	2	All	Yes	Nov. 17

Table 4. Sprint 3 Story Points

8.5.2 Retrospective

Over the course of this week our team has continued our work on the analyst platform. One aspect that we can do better on is with communication between the Head-Sponsor and the Data Analyst we were introduced to. Because we were only scheduled to meet once every two weeks with the Head-Sponsor, we tried to use other contacts to guide us in the correct direction. Although they have all been helpful, we ultimately needed to be more direct with questions and go to the person in charge, because he has the ultimate vision for the project, whereas our other contacts are there to answer specific questions. Another action to be taken is to improve communication of software descriptions and uses. We had a problem this week when one of our anomaly detection functions behaved in an unexpected way. The problem was caused by the operator believing that the function behaved in a certain way, as the updates made to it were not clearly communicated.

8.6 Sprint 3.5

8.6.1 Summary

This week was Thanksgiving week, so we worked on Monday and Tuesday. On Monday, we worked on figuring out the correct table to use for the TA data. We were able to obtain the correct table by searching different warehouses in Snowflake. On the same day, our bi-weekly meeting with our sponsors and analysts was canceled due to time conflict on their end. However, we did confirm that we are working in the correct table with our technical sponsor. On Tuesday, we took a closer look at the table with the goals of understanding what the table did and did not

contain, and of discovering how to map our anomaly detection functions onto the new structure. The task for this week was not completed as we did not fully understand every aspect in the new table and what other table we have to join it to in order to create the asset bridge.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Find the proper table in Snowflake for TA data	-Understand the columns -Determine what we should look at to find anomalies in new table -Run Queries	5	All	Yes	Nov. 28

Table 5. Sprint 3.5 Story Points

8.6.2 Retrospective

As this week was cut short due to the holidays, there is not much to report and improve upon. A small improvement could be keeping in contact with our tech sponsor and the analyst who previously built the FA bridge, just so we can have less blockers.

8.7 Sprint 4

8.7.1 Summary

During this week, we had a meeting with our technical sponsor. In this meeting we got more insight on the new table that we have received and were informed that we needed access to another table in order to compare starting and ending balances. We sent an email to the analyst that can give us access to this table. We also received another task for us to explore, which involved finding correlations in anomalies in order to determine why the anomaly happened. The ladder end of the week included finalizing the paper as well as exploring correlations in the asset bridge.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Understand and fix python code to fit the TA table	Reach out to analyst and our technical sponsor regarding the new table to understand the logic of the transactions and finding the anomalies in this paper	5	Austin, Nathan, Mitchell, Aruzhan	Yes	Nov. 30
Start drafting the final presentation to Sponsors	-Intro -Progress	3	Aruzhan	No	
Complete Final Paper Draft	Add missing sections in the paper Includes everything but the final week of programming	3	Nate, Mitchell, Aruzhan, Austin	Yes	Dec. 2
Feature Importance	Find importance features in Assets and Flows diagram using random forest	3	Aruzhan	Yes	Dec. 5

Table 6. Week 4 Story Points

8.7.2 Retrospective

This week was spent mainly in two areas. First the team began working with the new table we had been provided with, allowing us to work on the raw TA and FA data. A moment of reflection for here is that if we had more direct communication with our head sponsor, we could

have had access to this data earlier. The other area we focused on was the report, where we believe that our work has been satisfactory. Our previous efforts to put aside time each week for the paper has removed much of the potential stress of producing a draft this week.

8.8 Sprint 5

8.8.1 Summary

During this week, the team mostly worked on finalizing the paper for our draft on Friday. We met with our advisors several times this week to get insight on the edits we should be making. Along with the paper, the team continued to test out and implement the queries used to find anomalies in the TA data. Thursday, we met with our sponsor to get more into detail about the project risks and the overall risks associated with Fidelity. We also began planning out our final presentation and came up with a date of Wednesday the 14th to present to our sponsors and advisors.

Story	Subtasks	Points	Assignee(s)	Completed?	Date Completed
Polish and make edits to final paper	Review comments from the professors	5	Aruzhan, Mitchell, Nate, Austin	Yes	Dec. 9
Create query to mark anomalies in the TA data	Format the date column, check transactions with assets, compare day to day asset balances	4	Aruzhan, Mitchell, Nate, Austin	Yes	Dec. 9
Complete final presentation	Finish slides for presentation Run through mock presentation	4	Aruzhan, Mitchell, Nate, Austin	Not as of December 9th	

Table 7. Sprint 5 Story Points

8.8.2 Retrospective

At the beginning of this week, the team should have asked our advisors more specific questions about the paper. We had seen our professors' comments about different areas of the paper, but we had thought that they were minor problems, as opposed to tasks that could require more research and reflection and had to be completed as soon as possible. This led to the last days of the week, where although the team thought we were in solid shape for the paper, the team had to make major revisions. The team would have benefitted from spending more time reviewing comments from the professors and meeting with them to discuss them soon after they were posted, rather than delay them.

8.9 Burndown Chart

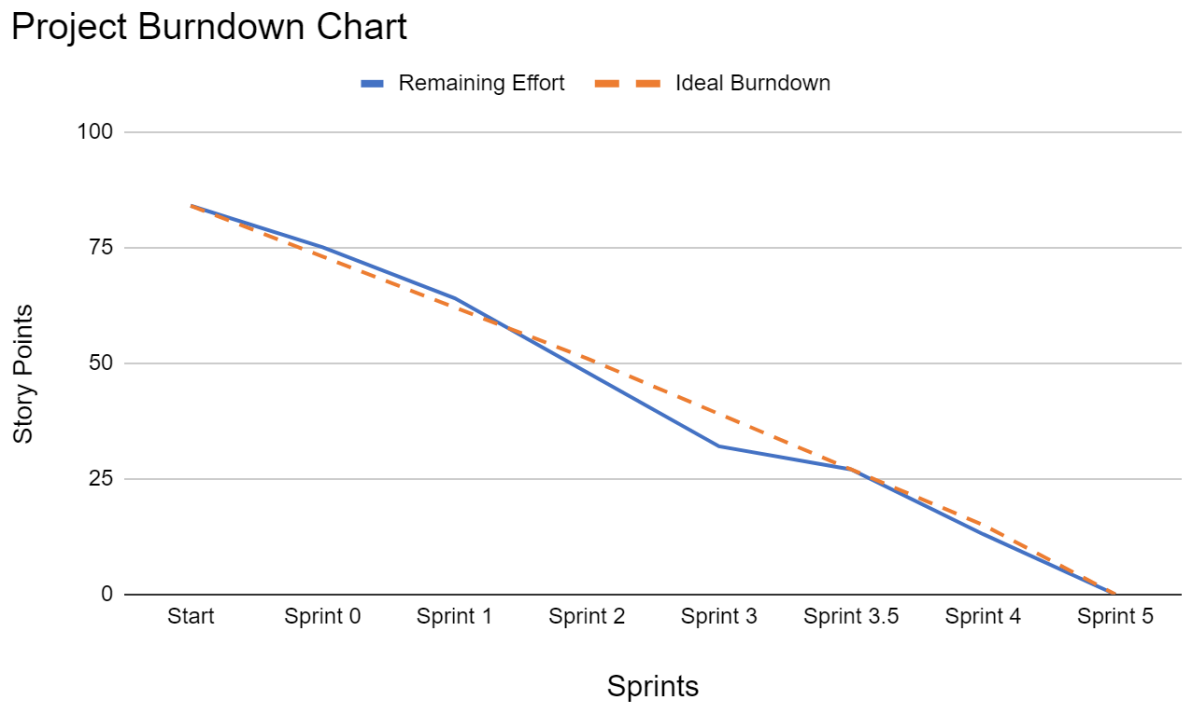


Figure 10. Sprints Burndown chart

Above is our team's burndown chart for this project. The project took place over 7 weeks, each of which was its own sprint. We began with a Sprint 0 as there were administration tasks and access requirements that were to be met prior to development, and so the first week was set aside for those tasks. In a similar fashion what would have been Sprint 4 was instead titled as 3.5, to designate that the sprint was a shorter one due to the holiday recess. Over the course of these 7 sprints, 84 points were assigned to our user stories, and the rate at which they were completed by the team may be seen with the solid blue line above. Similarly the ideal rate of completion is expressed by the segmented orange line. Their relative descents represent how close our team's actual progress was to our planned production. In this way it can be seen that early in the project the team began to fall behind our goals, before increasing the pace as we went into the holidays. After the holidays we remained close to the planned efforts until the end of the project.

9. Assessment

9.1 Goals Reached

The team created an anomaly detection model capable of notifying the company of errors within the TA/FA asset bridge data set. At the current state, the program is capable of detecting anomalies based on errors between the actual ending balances and the calculated ending balances for funds. Future steps will involve taking in larger samples of the TA/FA asset bridge data and using machine learning algorithms to identify correlations between variables in the data set and the anomalies.

The team was brought onto this project with the high-level goal of improving Fidelity's anomaly detection systems. In service of this goal, the team has worked to meet incremental progression goals that would support these efforts. Of these goals, the first the team reached was an understanding of the limitations of the current model as a tool for analysts. Upon this realization, the team worked towards the implementation of new models which would provide support to analysts as they patch anomalous data.

The featured TA, FA, and TA/FA dataset consist of hundreds of million data points. To clean the data, any singular instance that had more than 15% of its features having missing values were removed. For data instances that remained after the filtering that still had missing values, the unknown entries were labeled with a 0. This was a preventative measure for the anomalies that could have occurred, however in our case when working with Assets-Flow bridge and Consumption Account datasets there were no missing values. While working with a large dataset consisting of a wide variety of features, it was important to extract the most meaningful features using random forest (RF) feature importance. We trained eight regression models, each tasked with identifying the fund that resulted in the biggest anomaly.

Additionally, the team spent the last two weeks of the project beginning the construction of internal Snowflake queries and models which would work directly on the TA data. When this model is completed, it will provide Fidelity with a new layer of data integrity assurance, as catching anomalies in the initial TA data will prevent the possible downstream impact.

9.2 Learning Experience

Throughout the entire MQP experience, the entire group gained experience and enhanced our skills in the realms of technology and business. Working in a corporate environment gave us experience in tackling real life business problems and using real data. Additionally, we learned about the operations and expectations of working at a large financial company. This project gave us the opportunity to work with softwares, such as PowerBI and Snowflake. The team also gained an opportunity to implement the agile methodologies that we learned through classes and replicate them in a workplace setting. For some of us, it was also the first time entering and working within an office. This allowed us to see first hand what work life in our preferred fields will be like after graduation.

9.2.1 Technical Experience

Because our group consisted of different majors and backgrounds, we each benefited from the technical experience differently. We used a variety of different software platforms to help guide us through this project. When querying and analyzing the data for anomalies, we reinforced our SQL and Excel skills. We learned how to work with relational database systems and create new bridges/connections using primary and foreign keys.

This was all members' first time using Snowflake, and also our first time dealing with large databases. Working with Snowflake everyday has increased our confidence in being able to navigate big data, even in unfamiliar software environments. In PowerBI, we explored constructing various visuals, such as tables and dashboards. We created our own simple dashboards to understand some of the TA/FA data. Although it was not a deployable dashboard, we still gained basic skills and understanding of PowerBI. These are all industry standard tools, so it was great that we had the opportunity to pick some skills up using them. Jira is another industry standard tool that is used for project management. With Jira, the group learned how to create sprints, make a story, and assign points. We were also able to learn how to assign work and track progress with a burndown chart.

As industries continue to modernize, we see companies utilizing ever increasing amounts of data to run their businesses and generate profit. The team gained valuable experience in learning how companies manage and utilize large datasets. Our first piece of learning stemmed

from understanding the difference between an application testing (UAT) and production environment. While working with raw data, the team used the UAT environment on Snowflake. In this raw data, we noticed that some tables did not always have the best quality data, so we needed to be careful about reporting with that data. The data lacked in quality, because there was a lack of reporting rules, therefore making it seem unorganized. For example, in the raw TA data, there was no clear rule for why transactions were accounted for in the account balance for current day versus next day. In situations like this, we had to create our own reporting rules to work around creating accurate results. For the above example, we identified an anomaly if the sum of transactions did not match the current and next day balance, instead of just the current day. Because UAT is a test environment, developers may be prototyping new features, which can result in poor quality data. Second, we learned that gaining access to data can take multiple days. There were instances where we had to pause work on our project, because we did not have access to the correct data warehouse; that is something that we now know to account for future projects.

The last area of technical growth we experienced was working with relational database management systems (RDBMS) on large datasets. Relational databases are great for tasks of any complexity, so it's important to know how to get the best out of them. We learned the basic operations used in RDBMS like indexing, storing the dataset, using limits and count. Compared to a non-relational database management system, working with RDBMS results in faster execution of the sorting, inputting and searching for data. This makes a huge difference when working with large datasets like Fidelity TA/FA data. Using the company's limited virtual CPU, we had to make sure that our queries returned results within a reasonable time.

9.2.2 Business Experience

From a business strategy standpoint, the team learned about how large companies like Fidelity organize themselves. In a scrum environment, squads consist of a balance of people in different roles. In our liaison’s group, for example, they had data analysts, programmers, QAs, and scrum masters. We quickly learned that this type of group organization requires teamwork and communication. Nobody knows everything, so group members are constantly asking each other questions and helping each other solve any issues. Our team experienced this first hand with the many associates that we reached out to for help for the duration of our project. In many

instances, the situation would occur where we would be talking to one associate and they would recommend that we talk to another associate with more knowledge on our question. In this manner, we built a network of connections just through asking questions. Through this experience, we learned that well functioning businesses are interconnected and self supporting rather than segmented.

From a leadership perspective, we saw how managers delegated tasks for workers to figure out. Our project sponsor, for example, oversaw the project, but was not involved in the day-to-day activities of the project. Instead, he would do biweekly check-ins with the team to provide feedback and advice on the progress made. He also has a broader perspective of the entire business process, so he knew the right people to connect us to for further help with the project. In this sense, we learned that fundamentally managing is seeing the bigger picture of what is going on, then finding and connecting the right people to get tasks completed. While workers deal with the nitty gritty of tasks, managers help projects stay on task and give advice on how to remove any obstacles. We also had a chance to look at our project sponsor's calendar when scheduling meetings. Immediately, we noticed how busy his schedule was everyday. It seemed like almost every minute of every day was filled up with a meeting. This goes to show how communication is an essential skill for any manager to have. Seeing his busy schedule also made us much more conscious about how we were using our meeting time with him. We made sure to plan out our meetings and have any questions and roadblocks at the ready.

Working for a company allowed us to gain insight of what we should expect when we start working for a company and how to operate within a business. This type of experience is hard to replicate in a classroom setting. Everyone at Fidelity was accommodating and helpful, so we felt comfortable in this business environment. For the team members that have never been in a corporate setting before, it was a great learning experience on business professionalism and how people conduct themselves in a business setting. Just from the one visit to the office, we were able to see first hand what a work environment looked like and the opportunities available. This included simple things, for example: how to dress, how to deal with business scheduling, working with international workers, how to accommodate time, and how to ask for help.

When trying to schedule meetings in a busy environment, we had to learn quickly how to accommodate time for meetings. Often, we were working with international workers, so it was important to take into account different holidays and time differences when scheduling meetings.

Fidelity is a fast paced environment, where associates are always working on something. This made us quickly learn that it is important to ask for help early and often. It is uncommon for people to constantly check in on you, meaning that you have the responsibility of reaching out to get help. We had to strategize on how to utilize other people's time effectively, by knowing what specific questions to ask and how to ask those questions.

We met a lot of new people in a short amount of time. From this, we learned how important it was to carry yourself in order to make a good first impression. Collaboration is a big aspect in every industry, making it important to have a good relationship with your colleagues. Establishing trust and being accountable are two big factors when trying to leave an influence on your specific team or company. A small, but important learning to have, is that we learned that it is great to have an "elevator pitch" about yourself when meeting new people. For our team, we had a pitch that described who we were, what an MQP was, and how we were hoping to deliver value to the company. This quick 30 second introduction gave people an idea of why we were there and how they could better help us.

From an industry perspective, we learned about how technology was revolutionizing the finance industry. Our project liaison mentioned that about 15 years ago, Fidelity was more of a purely financial company. Now, Fidelity spends over $2.5 billion USD per year on technology (2018, Rooney). This shift towards the intersection of finance and technology can be seen through the adoption of data cloud services like Snowflake or through the use of AI/ML in projects like this one. The overarching theme is that technology can create more opportunities for value and efficiency within a business and the products that it delivers to its customers. For example, our project liaison talked about how once all financial data is migrated onto one centralized platform, he expects it will open a new path of opportunities for data analysis in the predictive and prescriptive realms. Right now, most data analysis is done in the descriptive and diagnostics realms. Analysts are only looking at what happened and why something happened. With projects like this one, Fidelity is moving towards predicting when something will happen in the future and knowing what the best course of action to take is. These types of analyses will be much more valuable to Fidelity and even other companies in the future.

10. Future Work

The team succeeded in laying the groundwork for the development of a platform for data analysts. Through the many aspects of our project, there are many opportunities for future teams to continue the work on a much more automated anomaly detection system for the TA and TA/FA data bridges. Future work would focus on two main areas:

1) Teams can use existing queries to identify anomalies in the TA data at the share quantity and values levels.
2) Teams can use our python program to run correlation tests between individual or multiple variables in the data and the anomalies.

10.1 TA Data

To produce a TA anomaly detection program, the continuing team can use the Snowflake queries that we created. In the queries, they will be able to identify any anomalies that occur at either the share quantity or value level. Transaction rows are automatically added to the asset rows. In this manner, any inconsistencies between changes in the shares quantity and transactions are identified. The team recommends using share quantity, because we noticed issues with inconsistencies in the data with the net asset values (NAVs) of shares. With the data we sampled (November 2022), the team noticed no issues with the data share wise. All transactions accurately depict changes in the asset balances.

The queries may prove sufficient in detecting any anomalies, and the team doubts that there are any issues with the transaction and assets data at the share quantity level. To fix issues with the NAV of shares, future teams will need to understand how and when the net asset value is calculated. With the raw TA data, there is no beginning of day balance recorded for the NAV. Therefore, to compare balances for a bridge, the end of day balance of the previous day is used to compare with the current end of day balance. At the share level, this works, because no trades happen during market close. For NAV, the price is constantly changing, so there is more room for error in the reporting of the data. Our team suggests that future work make the reporting rules for NAV clear, and create a solution around standardizing the recorded NAV across tables.

10.2 Asset Bridge

In the TA/FA asset bridge, the team recommends deploying a method to find why anomalies happen. The process would start with filtering out the table to only see anomalies, and then finding correlations between the anomalies. Some of these correlations could include specific days of the month, issues with certain flows, or reporting sources. For example, there is a possibility that an anomaly can occur because a specific aspect in the data flow increased or decreased a certain percentage in consecutive days. The team created Snowflake queries (see Appendix A) and a python program (see Appendix B) to identify anomalies in the fund balances. Users can set their own thresholds to create their own definition of an anomaly. In our activity, we used a 5% variance and one million dollar difference. The next step for future teams to take is using the program to gather multiple months or years worth of anomaly data and run tests with different variables to determine possible correlations to error sources.

10.3 Application

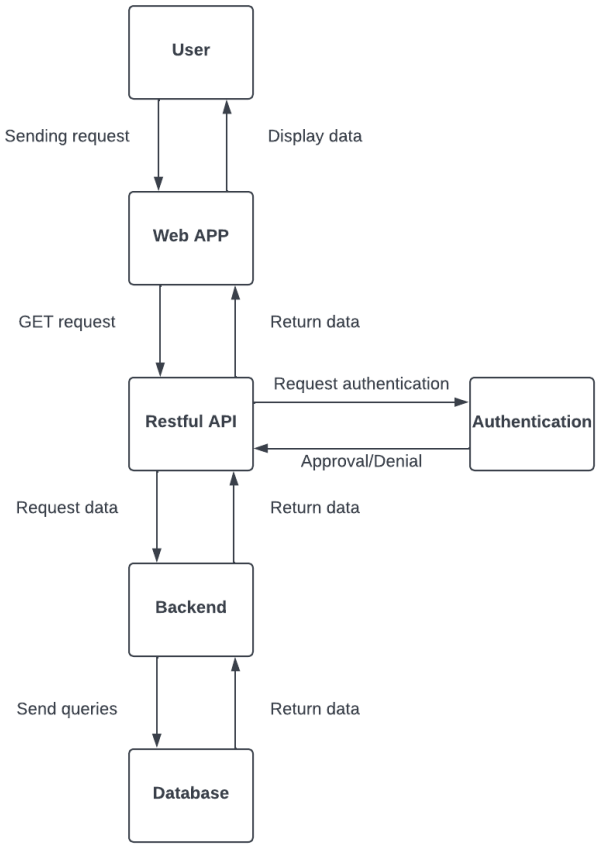


Figure 11. Architectural Design of Future Web App Implementation

The above diagram shows how data will be passed between different parts of the future web app implementation. First, the user will input their ID and request the data. The application will start compiling and send the request to the Restful API, which then will send an authentication request. If the request is approved, the Restful API will send a request to Backend to retrieve data from Database. If the request is denied, the Restful API will not make any calls and will stop any transactions.

11. Conclusion

At the culmination of this project, the team believes it to be a success. Over the course of this project the team has spent time mapping and prioritizing the flowing of data with Fidelity's systems. This data exploration was the dominant form of work during the early stages of the project, and allowed for the team to gain a better understanding of the characteristics of what is considered an anomaly to Fidelity. The team then focused on using those characteristics to identify anomalies as they appear; a process which was then automated by our first python model. This model was then expanded to provide the user with more customization options for finding anomalies at different thresholds of severity. The team then worked to apply new functionality to the model, centered around identifying features that are possible sources of anomalies.

This new functionality works to support analysts by reviewing flagged anomalies in search of possible causes. This searching took place on two levels for analysis, with the first being the aggregate flow level for accounts. This search investigated each sum of the flows for an account's daily transactions and searched for both commonalities between anomalies and flows that appeared to significantly contribute towards anomalies. Further investigation was also done at the individual transaction level for the funds, where all transactions would be considered to see how it contributed towards the anomaly.

The team concluded the project by turning towards the raw TA data for anomaly detection. As the TA data exists upstream of most other data we worked with, finding anomalies here is essential to ensure that downstream data would remain clean and reliable. For this data the team worked within Snowflake to create views which would allow analysts to quickly see when the shares within customer accounts are not being properly recorded or settled.

While this project is finished, the sponsors have shown similar enthusiasm for continuing this project with future teams (whether internal or external). In accordance with this decision, the team has also focused on identifying areas where there exists chances for future work, and documenting our findings for those who will continue the work of anomaly detection at Fidelity.

12. References

Alla, S., & Adari, S. K. (2019). *Beginning anomaly detection using python-based Deep Learning: With keras and Pytorch*. Apress.

Bajaj, A. (2022, December 7) *Anomaly Detection in Time Series*. MLOps blog. https://neptune.ai/blog/anomaly-detection-in-time-series

Biscobing, J. (2019, September). *Definition: Entity Relationship Diagram (ERD)*. TechTarget. https://www.techtarget.com/searchdatamanagement/definition/entity-relationship-diagram-ERD

Brunner, R.J. & Hariri, S. & Kind, M.C. (2021, April 1). Extended Isolation Forest. *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, 1479-1489, doi: 10.1109/TKDE.2019.2947676.

Cleveland, R. B., Cleveland, W. S., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics, 6*(1), 3. http://ezproxy.wpi.edu/login?url=https://www.proquest.com/scholarly-journals/stl-seasonal-trend-decomposition-procedure-based/docview/1266805989/se-2

Computer Science Center. (2018, August). Лекция 9. Прогнозирование на основе регрессионной модели [Lecture 9. Course 'Data analysis in python using examples and problem sets. Part1'][Video]. Youtube. https://www.youtube.com/watch?v=COBcXzKmOyk

Fidelity Investments. (2022). *About Fidelity - Our Company*. https://www.fidelity.com/about-fidelity/our-company

Fidelity Investments. (2022). *Quarterly Updates - Fidelity*. https://www.fidelity.com/about-fidelity/our-company/quarterly-updates/quarterlyupdates-q2-2022

Gunn, S. & Rogers, J. (2005). Identifying Feature Relevance Using a Random Forest Subspace, Latent Structure and Feature Selection. *SLSFS 2005. Lecture Notes in Computer Science, vol 3940.* Springer, Berlin, Heidelberg. https://doi.org/10.1007/11752790_12

Krishnan, A. (2019, March 3). *Anomaly Detection with Time Series Forecasting*. Medium. https://towardsdatascience.com/anomaly-detection-with-time-series-forecasting-c34c6d04b24a

Leffingwell, D. (2021, June 11) *Set-Based Design*. Scaled Agile Framework. https://www.scaledagileframework.com/set-based-design/

Liu, F.T.& Liu, T. & Zhi-Hua, K. & Zhi-Hua, Z. (2009). Isolation Forest. *Eighth IEEE International Conference*. 413 - 422. 10.1109/ICDM.2008.17. https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf?q=isolation-forest

Merryweather, E. (2022, February 10). *What Is the Spotify Model?*. Product School. https://productschool.com/blog/product-management-2/spotify-model-scaling-agile/

Mission Statement. (2022, April 1). *Fidelity Mission Statement 2022: Fidelity Mission & Vision Analysis*. Mission Statement https://mission-statement.com/fidelity/

Nau, R. (n.d.) *Notes on nonseasonal ARIMA models*. Fuqua School of Business, Duke University. https://people.duke.edu/%7Ernau/Notes_on_nonseasonal_ARIMA_models--Robert_Nau.pdf

Orac, R. (2019, September) *Time Series Prediction with LTSM*. Romanorac, https://romanorac.github.io/machine/learning/2019/09/27/time-series-prediction-with-lstm.html

Pedregosa F. & Varoquaux G. et al. (2011, October) Scikit-learn: Machine learning in python. *Journal of machine learning research, vol. 12, Oct, 2011.*

Peek, S. (2022, June 29). *What is Agile Scrum Methodology?*. Business New Daily. https://www.businessnewsdaily.com/4987-what-is-agile-scrum-methodology.html

Piikila, J. (n.d.) *What is SAFe?*. Atlassian. https://www.atlassian.com/agile/agile-at-scale/what-is-safe

Rooney, K. (2018, October 2). *72-Year-Old Fidelity Bets on the Future with Blockchain, Virtual Reality and Ai*. CNBC https://www.cnbc.com/2018/09/28/fidelity-the-tech-company.html

Scikit-Learn. (n.d.). *Sklearn.ensemble.isolationforest*. Retrieved November 12, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html

Snowflake. (n.d.). *Intro To Key Concepts*. https://docs.snowflake.com/en/user-guide/intro-key-concepts.html

Stadnik, V. (2022, November). *Segmented Linear Regressio*n. Code Project. https://www.codeproject.com/Articles/5282014/Segmented-Linear-Regression

Schwaber, K. & Sutherland, J. (n.d.) *Scrum Guide*. https://scrumguides.org/scrum-guide.html

Weiss, M. (2007, November 3). Fidelity Reorganization Will Cut Its Taxes. *The Worcester*

Telegram & Gazette.
https://www.telegram.com/story/news/local/north/2007/11/03/fidelity-reorganization-will-cut-its/52751677007/

13. Appendices

13.1 Appendix A- Python Code

```
[5]: bridge['Expected Ending'] = bridge['SUM(BEGINNING_ASSETS)'] + sum_of_flows
     bridge['Dollar Difference'] = abs(bridge['SUM(ENDING_ASSETS)'] - bridge["Expected Ending"])
     bridge['Percent Difference'] = bridge['Dollar Difference'] / bridge['SUM(ENDING_ASSETS)']

[6]: dollar_threshold=1000000
     percent_threshold=0.05
     bridge['Dollar Threshold Reached'] = ['Y' if x > dollar_threshold else 'N' for x in bridge['Dollar Difference']]
     bridge['Percent Threshold Reached'] = ['Y' if x > percent_threshold else 'N' for x in bridge['Percent Difference']]
```

Figure 12: Python Code

The first steps of our python program include calculating the expected ending of the assets, as well as the dollar difference and percent difference. The differences were calculated by finding how far off the actual ending assets were from the expected ending assets. A temporary threshold was set on the dollar and percent difference of $1,000,000 and 5%. The table will then make a new column in the table to display whether or not the threshold was reached (see figure below).

Expected Ending	Dollar Difference	Percent Difference	Dollar Threshold Reached	Percent Threshold Reached
4.524308e+09	2.293834e+09	1.028406	Y	Y
2.676524e+10	1.360965e+09	0.053572	Y	Y
1.884249e+09	9.421244e+08	1.000000	Y	Y
5.555982e+08	3.238883e+08	1.397819	Y	Y
3.328808e+08	1.907172e+08	1.341532	Y	Y

Figure 13: Dollar and Percent Thresholds

From here, we can then filter out the data to display only the rows that reach the threshold. This will result in the anomalies that we care about the most. To find the contributing factors in the data flow that caused this anomaly, we find the column that had the closest number to the dollar difference. The resulting column name was considered the contributing factor for the anomaly.

13.2 Appendix B- SQL Query for TA Data

```
FROM (SELECT TO_DATE(CONCAT(SUBSTRING(DAY_KEY, 0, 4), '-',SUBSTRING(DAY_KEY, 5, 2),'-',
                            SUBSTRING(DAY_KEY, 7, 2))) AS TDATE,
      CASE WHEN(DAYOFWEEK(TDATE - 1) = 0) THEN (TDATE - 3)
        WHEN (DAYOFWEEK(TDATE - 1) != 0) THEN (TDATE - 1) END AS TPREV_RECORDED_DATE,
      *
      FROM
        WHERE (YEAR(TDATE) =  AND MONTH(TDATE) = 11 AND DAY(TDATE) = 3)
        AND CUSTOMER_ACCOUNT_BK =
        AND (MEASURE_TYPE_DESC = 'ASSETS')) AS A
INNER JOIN (SELECT TO_DATE(CONCAT(SUBSTRING(DAY_KEY, 0, 4), '-',SUBSTRING(DAY_KEY, 5, 2),'-',
                                  SUBSTRING(DAY_KEY, 7, 2))) AS TDATE,
      *
      FROM
        WHERE (YEAR(TDATE) = 2022 AND MONTH(TDATE) = 11)
        AND CUSTOMER_ACCOUNT_BK =
        AND (MEASURE_TYPE_DESC = 'ASSETS')) AS B
ON (A.TPREV_RECORDED_DATE = B.TDATE
AND A.PRODUCT_BK = B.PRODUCT_BK
AND A.CUSTOMER_ACCOUNT_BK = B.CUSTOMER_ACCOUNT_BK
AND A.MEASURE_TYPE_DESC = B.MEASURE_TYPE_DESC)
ORDER BY DATE, A.MEASURE_TYPE_DESC
```

Figure 14: Raw TA Query

This query for the raw TA data table groups all assets holding for an account to a specific day. There is an added column that shows the account asset holdings for the previous day. In this manner, you can compare the current and previous day holdings. The next step for this query is to add transactions and add a column that identifies if the previous and current day holdings do not align with the transactions accounted for. This column identifies the anomalies within the data and by filtering with it accounts with anomalous totals may be detected.