Fidelity: Customer Data Exploration and Analysis with a Geo-Spatial Focus

Project Team

Janette Jerusal, jjerusal@wpi.edu

Jack Lafond, jwlafond@wpi.edu

Sandra Phan, sphan2@wpi.edu

Project Advisors

Professor Robert Sarnie
Professor Jim Ryan
WPI Business School

Professor Marcel Blais
Department of Mathematical Sciences/Data Science

Professor Wilson Wong
Department of Computer Science

This report represents the work of WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, please see http://www.wpi.edu/academics/ugradstudies/project-learning.html

Abstract

In collaboration with Fidelity Investments, this MQP explores their newly developed Business Performance Measurement dataset with customer level attribution. Our team created an application featuring an interactive map, data visualizations, and regression analyses, to uncover its key business insights including interactivity for extensive drill down capabilities, allowing the firm to understand their performance and customer tendencies in regions across the United States. This application serves as a prototype to understand the dataset's current capabilities and how they can be enhanced. This project also serves as a framework to guide future development, as its design is adaptable and can be updated to reflect current user needs or interests.

Executive Summary

Our collaboration with Fidelity Investments led us to understand that products and services can only be effectively marketed if they target the correct target audience. With this in mind, our MQP aimed to create a comprehensive customer map, highlighting where customers are located at state, county, and zip code granular levels. Additionally, their key demographics are included as well to help the firm identify what backgrounds their target audiences are from.

To create this map, we utilized an Agile approach to develop our project in alignment with the Scrum framework, which allows us to incorporate Agile principles within our work. Work was sectioned into sprints, where the team members would assign each other tasks to be completed by the next sprint. A definition of done was instantiated to serve as a baseline for determining when a task can be considered completed. Sprints contained Scrum meetings that occurred on a daily basis to serve as a time for each member to share their progress or any struggles they were facing. While our Sprint meetings served as valuable time to regroup, we found Jira to be a far more organized approach to staying on track with our tasks. Jira, a project management software platform, allowed us to allocate tasks into epics, a format in which tasks can be grouped together through shared similarities or priorities.

Our team utilized Visual Studio Code as this source code editor was light weight but featured a large extension library for all our needs. To create our interactive map, we utilized the Streamlit-Folium framework, a publicly available Python package. Streamlit allowed us to quickly set up a web-based application, and its connection to Python made it a great option to explore and visualize the dataset. The prototype we were given featured an interactive map with the capability to view the aggregated dataset at the state and zip code level. We were tasked with bringing these current capabilities into one seamless map and expanding them to create a complete and coherent data exploration application. In total we expanded the map to include much more interactivity with functionality to drill down from state to county, to zip, and back up all within one map. We also built out an informative tabbing feature where we hosted key information like aggregated data relevant to the geography currently selected, customer demographic visualizations, and Line-of-Business visualizations. Lastly, we performed

regression analyses to understand how customer proximity to Fidelity investor centers can impact business performance.

Incrementally building this map resulted in a need to find a way to demonstrate our weekly progress to the necessary stakeholders in an effective way, which is why we turned to Figma and Power BI to create mockup designs of our potential updates to the map. This allowed us to build our skills in communicating our ideas and presenting them in a constructive manner.

As a whole, this collaboration with Fidelity Investments was a valuable experience as it allowed us to approach one problem, creating an effective medium to demonstrate customer data, in many different disciplines – including software engineering, data science, and management.

Acknowledgements

We would like to first thank our advisors for their guidance and support throughout the term; thank you to Professor Sarnie, Professor Blais, Professor Ryan, and Professor Wong. We would also like to thank Marc Trudeau for his guidance throughout the term learning and navigating the Agile framework.

To our Fidelity sponsors, thank you for this opportunity, and for your help throughout this project. Thank you, David Wolf, Vivanne Luk, Mohanapriya Gnanamani, Sai Pranav Yarlagadda, and Abhisheg Selvaraj. Everyone at Fidelity was incredibly supportive throughout our project.

Table of Contents

List of Figures

List of Tables

1. Introduction

Fidelity is a global company with millions of customers. As part of the Business Performance Measurement Initiative, the firm-wide assets and flows information hosted in Finance Data Lake has been enriched with standardized customer level attributions. With this core data asset, the Financial Management Technology group within Fidelity has built a Geo Spatial Map framework using Python. In its current state, the map visual delivers functionality to explore assets and customer counts at the state and zip code grain levels. The goal of this project is to build upon Fidelity's current prototype, which includes an interactive map, inclusion of more datasets, abilities to partition data, abilities to flag different locations based on search criteria, and inclusion of time series analyses. By further developing this prototype we aim to help Fidelity employees discover key insights and make guided business decisions.

2. Research

2.1 Company Background

Fidelity is a major brokerage firm with the mission of helping the clients gain financial well-being. This service includes individuals, as well as companies aiming to deliver benefits to their employees. Currently, Fidelity manages over 43 million accounts and has $11.7 trillion in assets under management (Fidelity, n.d.). As of their 2022 annual reports, Fidelity produced $25.2 billion in revenue and had an operational income of $8 billion. They are headquartered in Boston, MA, and currently employ about 70,000 people globally (Fidelity, 2023).

When approaching this project, we wanted to ensure that we were allowing a logical continuation to exist between previous collaborations with Fidelity Investments. With that in mind, we spoke to a member of the 2022 Fidelity CTG team to not only understand the scope of their work with the company, but to understand how their team operated as a functional unit as well. While they did mention that a large focus of their project was in terms of manipulating the data through software engineering, there was less of a focus on a visual aspect of their project –

such as the web application we intended to build. This allowed us to proceed with our original idea, seeing as it seemed like a further expansion of the work conducted by the 2022 team while also expanding on our own skills.

2.2 Regressions

Regression analyses seek to quantify the relationship between a dependent variable, and independent variables, also called explanatory variables. This is often done through creating a line of best fit, which is a function that computes a prediction for an independent variable, based on inputs from the dependent variables (Gallo, 2015). There are many types of regressions that help to identify correlations between variables, for our project we used two, linear regressions, and polynomial regressions.

2.2.1 Linear Regressions

A simple linear regression assumes that the relationship between a dependent variable and one independent variable is approximately linear (James et al., 2017). The simple linear regression equation is shown below:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x,$$

Figure 1 – Simple Linear Regression Formula

Where ˆy indicates a prediction of Y on the basis of X = x. Here we use a hat symbol, ˆ , to denote the estimated value for an unknown parameter or coefficient, or to denote the predicted value of the response. (James et al., 2017)

A dataset is used to calculate these coefficients, where their estimates provide the line that most accurately predicts the dependent variables. In order to do this an error function called the residual sum of squares (RSS) is minimized. Each ith residual is the difference between the actual ith response value and the predicted ith response value (James et al., 2017).

$$\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2,$$

Figure 2 – Residual Sum of Squares Formula

Where $e_i = y_i - \hat{y}_i$ represents the ith residual—this is the difference between residual the ith observed response value and the ith response value that is predicted by our linear model. (James et al., 2017)

If $\bar{x}$ and $\bar{y}$ are defined as the sample means or, $\frac{1}{n}\sum_{i=1}^{n} x_i$, and $\frac{1}{n}\sum_{i=1}^{n} y_i$, respectively, then with calculus the coefficients that minimize this function can be determined as:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x},$$

Figure 3 – Linear Regression Coefficient Calculations
(James et al., 2017)

After the regression is created it is important to be able to evaluate its performance. One common calculation used to understand model performance is the R^2 statistic. This statistic measures the proportion of variance in the response variable that can be explained by the independent variable (James et al., 2017). The formula for calculating the R^2 statistic is shown below:

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

Figure 4 – R-Squared Formula

Where TSS = $\sum(y_i - \bar{y})^2$ is the total sum of squares. (James et al., 2017)

The R^2 statistic will always be between 0 and 1, and a statistic closer to 1 indicates a stronger relationship between the dependent and independent variable. However, as highlighted in An Introduction to Statistical Learning, this measure is relevant to the field that it is being applied to; "[Regarding R-Squared] in typical applications in biology, psychology, marketing, and other domains, the linear model is at best an extremely rough approximation to the data, and residual errors due to other unmeasured factors are often very large. In this setting, we would expect only a very small proportion of the variance in the response to be explained by the predictor, and an R^2 value well below 0.1 might be more realistic" (James et al., 2017).

2.2.2 Polynomial Regressions

Polynomial regressions extend the linear model by including higher powers for the independent variable. This allows for the modeling of nonlinear relationships such as quadratic or cubic relationships between the independent variable and the response variable. Typically, degrees higher than the 4th degree are avoided as they can cause overfitting. These models can still be evaluated using the R-squared statistic (James et al., 2017). The formula for a polynomial regression of degree d is shown below:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_d x_i^d + \epsilon_i,$$

Figure 5 – Polynomial Regression Formula

Where $\in_i$ is the error term. (James et al., 2017)

3. Methodology

Agile methodologies stood as our guiding principles that reshaped the way teams collaborate, innovate, and adapt in the environment of project management. Agile's emphasis on flexibility empowers teams to pivot swiftly, ensuring projects remain relevant amidst changing landscapes. This adaptability not only guarantees the end results align seamlessly with stakeholders' ever-evolving needs but also fosters a profound sense of satisfaction and relevance in the final deliverables. This methodology was created as a result of failed waterfall projects in the past (Atlassian, n.d.-b).

Transparent communication and clearly defined roles are the keystones of success within the Agile framework. Specifically, this project applied the Scrum framework. This framework features structured events such as Sprint Planning and Daily Scrums that foster open communication channels, which enrich collaboration among team members. Roles like Scrum Master, Product Owner, and Developers bring clarity to responsibilities, reducing confusion and nurturing a sense of shared accountability. This transparent collaboration not only streamlines workflows but also enhances the overall synergy of the team, ensuring a harmonious blend of expertise and creativity. Periods of work in Scrum are sectioned by sprints. During a sprint team members assign themselves tasks from the sprint backlog and complete them according to the team's definition of done. The sprint ends after the allotted period, and any unfinished tasks are put back into the product backlog. The product backlog is the prioritized list of work for the product, determined by the project's requirements. This backlog can change over the course of the project as customers’ needs change or the views of the Scrum team change (Atlassian, n.d.-c).

Sprint backlogs are created during sprint planning meetings where members of the team decide what work, from the product backlog, to include in a sprint in order to meet a sprint goal defined by the product owner. At the end of each sprint, the team holds a sprint review where each member goes over the work they have done and run informal demos. During this process, feedback is given and new ideas for the product can form. The team also holds sprint retrospectives with the purpose of evaluating how the team is carrying out their Agile Scrum process. During these meetings the team identifies areas of improvement in order to become a more cohesive group. In order to properly practice Scrum, we will use Jira, a project management software, that allows us to track, update, and review the project through Scrum structures.

In the realm of data analysis, Agile methodologies amplify the impact of data-driven decision-making. Techniques such as machine learning and anomaly detection empower organizations to extract meaningful insights from complex datasets. By leveraging these methodologies, businesses gain a competitive edge, armed with the knowledge to make informed, strategic decisions. The iterative nature of Agile aligns seamlessly with the iterative process of data analysis, refining insights over time and leading to more accurate and valuable conclusions.

Effective implementation of Agile requires a customized approach tailored to the unique DNA of each organization. Understanding the specific intricacies of an organization's culture and workflows allows Agile methodologies to be seamlessly integrated. This tailored approach enhances operational efficiency, ensuring Agile principles align harmoniously with existing structures and processes. Comprehensive training programs are pivotal in empowering teams with the skills needed to navigate the complexities of Agile methodologies. A well-trained team becomes agile, capable of adapting to challenges with grace and precision. Continuous training and upskilling foster a culture of innovation, ensuring teams remain at the forefront of Agile practices.

Agile methods thrive in technical environments, fostering continuous growth and learning. Regular and post feedback loops serve as invaluable tools, enabling teams to learn from both successes and failures. These practices identify areas for improvement. Thus, the organization is able to improve its efficiency. This culture of iterative improvement ensures that Agile methodologies are not merely a one-time implementation but a continuous evolution, adapting to the ever-changing demands of the business landscape.

In the intricate tapestry of Agile methodologies, adaptability, transparent communication, and data-driven insights intricately weave together a narrative of success. Implementing Agile transcends being a mere process; it embodies a mindset transforming how projects are managed and data is analyzed. It empowers teams to face the unknown with confidence, arming them with tools and strategies necessary to enter in an ever-evolving world. Embracing the Agile spirit marks the beginning of a transformative journey, where collaboration, innovation, and adaptability become not just goals but ingrained values. Agile methodologies propel organizations toward a future of unparalleled success and resilience.

Our technological journey also demanded mastery of Power BI and Python. Power BI, with its dynamic visualization capabilities, brought our raw data to life, enabling us to craft compelling narratives and actionable insights. Python, our chosen coding language, facilitated the implementation of adaptive machine learning models, ensuring our analyses remained responsive to evolving data patterns.

Risk management stood as our guiding light in the intricate landscape of corporate software development. Operational risks emphasized rigorous testing and streamlined communication. Within the domain of system integration environments, we found a secure space for experimentation, which fosters innovation without disrupting critical data flows. Financial risks loomed as a constant reminder, underscoring the importance of early difference detection and error identification, safeguarding the integrity of financial statements. Reputational risks highlighted the delicate balance between technological innovation and customer trust, urging stringent quality assurance and unwavering ethical commitment. This responsiveness ensured the reliability of our solutions and encouraged Fidelity's reputation as a forward-thinking and trustworthy partner.

4. Software Development Environment

4.1 Project Management Software

4.1.1 Access Software

Our team used Citrix Desktop, a remote desktop application that allowed us to access our Fidelity virtual machines remotely. Each team member was assigned a virtual desktop. However, some team members had technological issues with the remote desktops and were provided with company laptops. All programming and communications with the sponsors were done through these virtual machines and company laptops.

4.1.2 Microsoft Teams

To communicate with our sponsors, we followed Fidelity's standard of using Microsoft Teams. This software allowed us to communicate quickly with our sponsors, which included scheduling meetings, asking project-related questions, and sharing progress with the project.

4.1.3 Jira

To adhere to the agile framework the team used Jira from Atlassian. Jira is a project management software that allows for centralized project tracking (Atlassian, n.d.-a). Using this software, we were able to create our epics, organize a product backlog, plan out weekly sprints, and evaluate our progress. During each sprint tasks from our backlog were brought into our current sprint. Each team member could see which tasks were either To Do, In Progress, or Done as well as which team member was working on different tasks. Jira also provided various reports which allowed our team to reflect on our process each week and improve our methods.

4.2 Programming Environment

4.2.1 Visual Studio Code

Visual Studio Code (VS Code) is a lightweight source code editor made by Microsoft. This editor can run on Windows, macOS, and Linux systems. VS Code includes a large extension library that can enhance its features and make it adaptable to a variety of needs (Microsoft, n.d.). We used VS Code to develop all our python scripts for this project.

4.2.2 Python

Python is an interpreted, object-oriented, programming language. Python is used for many scientific applications as it is quick to set up, and has an extensive library for mathematical computations, data processing, and visualizations. As a result, python is a preferred choice in the data science industry (Python Software Foundation, n.d.).

4.3 Data Sources

Fidelity's Business Performance Measurement data (BPM data) is stored on Snowflake. Snowflake is a cloud-based data warehouse that is easily scalable allowing for more efficient loading, integration, and analysis of data (Snowflake Inc., n.d.). For security reasons, our team was not able to access Fidelity's BPM data through snowflake. Instead, we were provided anonymized and aggregated data through our Fidelity sponsors in the form of comma-separated values (csv) files.

4.3.1 Pandas

Pandas is a python library that aims to assist developers in data loading, processing, and analyzing. We made use of Pandas' DataFrame objects to load the BPM data from the csvs into our scripts and for further processing. This library allowed our team to quickly visualize missing, or corrupted data that we could then clean from our datasets. Pandas also features functionality for aggregation, and table joins that was utilized by our team to group our datasets on different attributes and include important information from other data sources.

5. Software Requirements

5.1 Requirements Gathering

We used our PQP as time to meet with our Fidelity sponsors and outline the project. During this time, we became familiar with the business, the project, and the current prototype. We were also given a project outline to help map our journey and track our progress. This allowed us to create a backlog and plan out the development that would need to be done. After getting familiar with the company and the project, we set up weekly meetings with our Fidelity contacts to track progress with them. In these meetings we showcased the tasks that had been accomplished in our previous sprint, and our plans moving forward. This gave our sponsors an opportunity to review our work frequently and make suggestions or adjustments to our plans as they saw fit. Through this process we were able to constantly refine our software requirements, ensuring that we were delivering a valuable product.

5.2 Functional and Non-Functional Requirements

Our team was tasked with creating a data exploration and visualization application with a focus on geospatial analytics. This application's purpose was to explore the possibilities of the dataset created by the Business Performance Measurement Initiative in uncovering important insights. The prototype we received featured an interactive map which over the course of the MQP we would improve and add to. The following table highlights the requirements for our application.

Functional Requirements	Non-Functional Requirements
Ability to explore US geographies at different grain levels like State, County, and Zip	Layout of the application is clear and easy to navigate
Displaying of key performance metrics for these geographies	Exploring the map is efficient
Visualizations highlighting insights for customer demographics and LOB Groups	
Incorporation of other datasets like population data, and investor center proximity data	

Table 1 – Functional and non-functional requirements for application

5.3 Epics and User Stories

Using these requirements and through our weekly meetings with our Fidelity sponsors, we created the following epics and user stories for our project which guided our development. The three tables below will explain what our epics are as well as the goal and user stories.

EPIC 1: SETUP PROJECT	
GOAL	Get all necessary systems running so that we can start to develop the application
USER STORIES	- Determine Project Details - Setup Project Software (Jira, IDE, etc.) - Get Prototype Running

Table 2 – Epic 1: Setup Project

EPIC 2: GEOSPATIAL ANALYTICS	
GOAL	Provide valuable insights based on the geospatial and demographic attributes of the data
USER STORIES	- Include tabulated data relevant to map selections - Create visualizations for current geography LOBs - Create visualizations for current geography demographics - Investigate relationships between attributes - Include other relevant datasets to enhance insights

Table 3 – Epic 2: Geospatial Analytics

EPIC 3: INTERACTIVE MAP	
GOAL	Develop a seamless interactive map for data exploration across geographies
USER STORIES	- Allow for drill down from state to county, to zip - Include on click interactivity - Include ability to search for geographies - Incorporate heatmapping on multiple attributes

Table 4 – Epic 3: Interactive Map

6. Design

6.1 Streamlit

We developed our app within the Streamlit framework. Streamlit is a free, open-source framework that allows users to quickly build data science web apps using only python (Streamlit Inc., n.d.). Streamlit's API allows for web structures, widgets, and other features to be implemented rapidly and easily. As we built out our product, we used a multipage design where we could develop different maps and experiment with new features. Each page is its own python script and the Streamlit framework allows for two-way binding and caching without requiring users to implement html or JavaScript. The following figure is Streamlit's flow model for a web app.

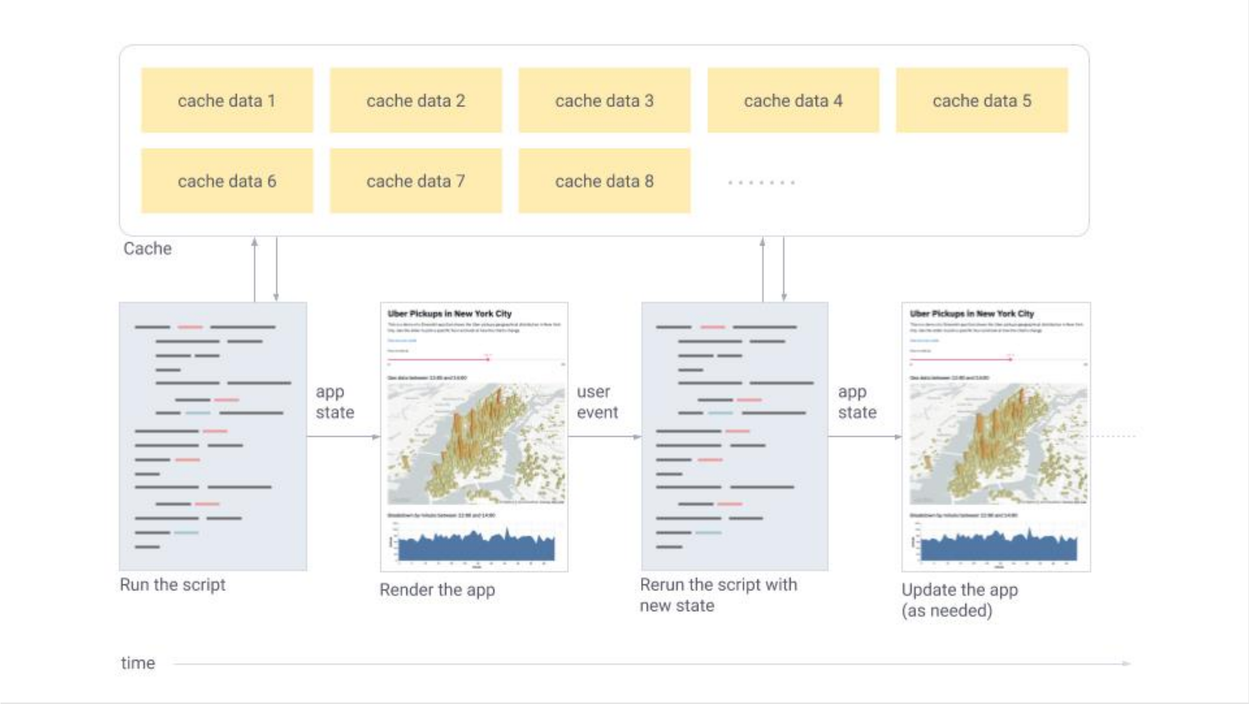

Figure 6 - Streamlit Web App Flow
(Streamlit Inc., n.d.)

6.2 Mockups

We used Figma to plan out our designs and interactivity implementation. Our initial Figma mockups are shown below:

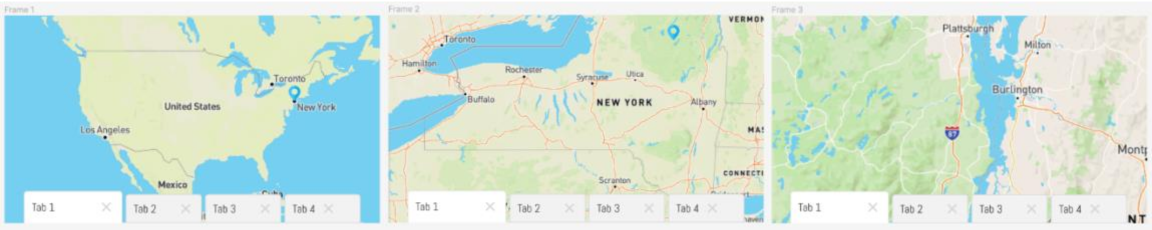

Figure 7 – Initial Figma Mockups

These images show our initial plan to drill down from the total US view to a state and its counties, and then into a county's zip codes. These mockups also demonstrate our plan to implement a tabbing feature where each tab would store different information corresponding to the current geographical location selected.

7. Software Development

As mentioned, our development process utilizes the Agile framework. Jack served as the product owner, and Janette served as the Scrum Master. Each member also served as a developer. We had weeklong sprints over the course of the term which totaled seven weeks. We had daily scrums at 1pm over zoom and had regular meetings with our WPI advisors at the start of our week on Monday at 1pm. We also had weekly meetings with our Fidelity advisors to go over our progress, which were held on Wednesdays at 11am. On Monday's we took more time in our meetings for our sprint planning session, and Fridays we took more time for our review and retrospectives of that week's sprint.

7.1 (PQP) Sprint 0: 8/24 - 10/13

Our PQP acted as our sprint 0. During the PQP, we met with our sponsors at Fidelity, gathered project requirements, and gained access to our virtual machines. We also took time to study Agile, making sure that we were well prepared to start development at the start of B-term (10/23). Along with this we gained familiarity with Jira software, creating a project, product backlog, and preparing to start our first sprint.

7.2 Sprint 1: 10/23 - 10/30

7.2.1 Summary

Our first sprint began Monday, October 23rd. In our first sprint the team had a goal of getting all necessary systems set up and implementing some improvements in the software to show that all systems were working. We did run into some technological issues, as Sandra could not use the Fidelity virtual machine from her device, and Janette's virtual machine did not have Microsoft's software packages installed. After some meetings and emails with Fidelity, those issues were resolved, and the team should be able to fully start development next week. Jack's virtual machine was working, and he was able to iterate on the product. We were able to switch the prototype to a csv connection, and we were able to implement a select state feature that allowed the user to split a state into the zip grain level within the same map. While some technological factors held us back on development, the team was still able to be productive, as we revised this report, and Professor Blais provided us with access to his lectures on return modeling. These lectures will help us later in development when we include financial analysis on Fidelity's customer base. In total we were able to complete 18 of 19 story points. Our last story point was getting GitHub setup, which we are waiting for our Fidelity advisors to provide us with access to GitHub on our virtual machines.

User Story	Subtasks	Points	Assignees	Status
Research Return Modeling	- Study Professor Blais' lectures on modeling financial returns	3	All	Done
Revise MQP Report	- Update project description and software requirements	3	All	Done
Get Project Running and Get Familiar with Code	- Install VS Code to virtual machines - Install Python to virtual machines - Access source code - Take time to understand current code	3	All	Done
Install Python Packages	- Create a virtual environment for the project - Install current packages to the environment	1	All	Done
Seamless Drill Down	- Switch connections to CSVs	8	Jack	Done

	- Allow selection of a state and split that state into zip map			
Setup GitHub	- Gain access to GitHub repo once set up by Fidelity - Install git on our virtual machines	1	All	In progress

Table 5 – Sprint 1 User Stories

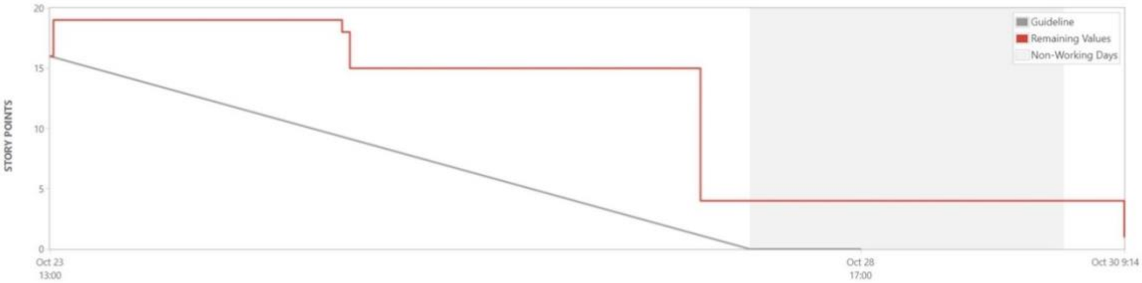

Figure 8 – Sprint 1 Burndown Chart

7.2.2 Retrospective

This week came with some challenges, as Sandra and Janette could not begin development due to technology issues. However, we were able to adapt well, finding other useful work to be done, such as Professor Blais' modules, and revising the MQP report. We also were able to share code over zoom and do pair programming. Given the circumstances, this week was still productive as we iterated on the product and completed a good number of story points.

Our daily scrum time of 1pm also worked nicely as it gave our team members a chance to start work before the meeting, and come to it with any blockers or questions, making the scrums more valuable. Also, our WPI advisor meeting on Monday prepared us well for our Wednesday meeting with our Fidelity sponsors. We plan on maintaining these meeting times.

For next week we hope to be able to have the team start fully developing. This should also mean an increase in story points that we can accomplish, and a bigger iteration of our project.

7.3 Sprint 2: 10/30 - 11/6

7.3.1 Summary

Our second sprint started on Monday, October 30th. We had a goal this week of improving the map functionality and refining the current state of the application. Sandra was able to get a laptop, however Janette was unable to get approved for one this week. Janette planned on picking up her laptop the following week when we visited Fidelity's Boston office for connect week. Another issue that arose was that the Streamlit package we are using is pending an important bug fix. This bug meant that the dynamic maps we were developing could not be styled correctly. Our plan adjusted to instead create two maps, one static map that re-rendered but accurately showed our heat mapping, and another map that we were unable to style but showcased our map interactivity. We also met with one of Fidelity's UX designers, who helped us understand more user-friendly practices for our application and helped us define some new user stories for our backlog. We also realized that our detailed pop-up story was no longer relevant as we adjusted our product backlog to reflect the use of tabs which we will explain further in the next sprint. The GitHub setup also continued into this sprint as we were unable to progress on it. Due to these challenges and changes we were only able to complete 6 of 20 story points, 11 of which will be carried into the next sprint.

User Story	Subtasks	Points	Assignees	Status
Clean Formatting	- Add a string to data frame that is a prettified version of the asset number - Change asset references to use this string	2	Jack	Done
Add Percentage Map	- Add a colormap based on percentage of customers to population in each state	2	Jack	Done
Optimize CSV Read	- Use Streamlit caching - Reference already created caches when creating new filtered data frames	2	Jack	Done

	- Research fastest methods for reading CSVs			
Click Through Drill Down	- Get zip data - Create variables to track selections and current map parameters - Split a selected state into zips	5	All	In Progress
Zoom In Drill Down	- Switch connections to CSVs - Allow selection of a state and split that state into zip map	5	All	In Progress
Setup GitHub	- Gain access to GitHub repo once set up by fidelity	1	All	In progress
Detailed Popup	- Gather all state and zip data - Attach popup to side of map and include relevant details	3	All	No Longer A Story

Table 6 – Sprint 2 User Stories

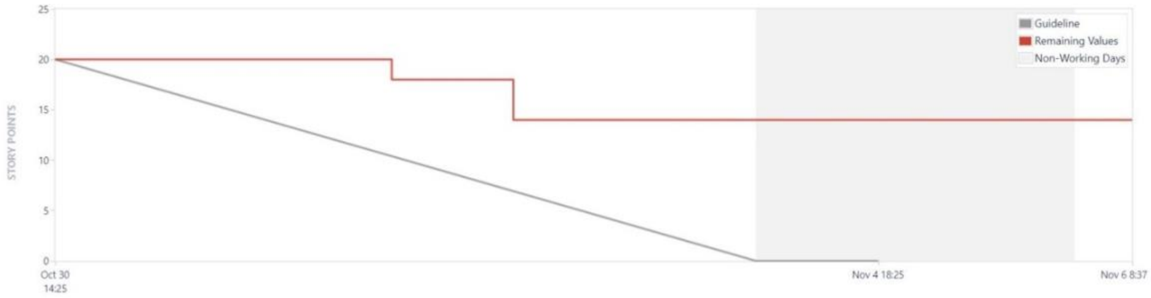

Figure 9 – Sprint 2 Burndown Chart

7.3.2 Retrospective

Some technological challenges persisted again through this week, limiting the team's ability to complete the full sprint. Issues outside of our control like bugs in Streamlit's library also held us back but we were able to pivot and find solutions in the meantime. Although 11 points are being carried over into the next sprint, a good portion of those stories have already been completed. For example, the map interactions are almost done, just a few things need to be fixed in the code. Likewise, our GitHub accounts have now been created but we still need access to the repositories.

Similar to last week, we have found that our Advisor meetings at the start of the week have put us in good positions to maximize the value of our Fidelity meetings later in the week. By voicing concerns and providing detailed updates our advisors can help us plan the best course of action with our Fidelity contacts. We have also become more acclimated to the time zone difference between our team and some of our Fidelity contacts. This week when we had issues that involved our contacts in India, we compiled our topics into a list and provided it at the end of the day to them. The following days we would wake up to a response and often an issue being fixed. These processes allowed us to be productive throughout our workday and avoid losing entire workdays from any blockers.

7.4 Sprint 3: 11/6 - 11/13

7.4.1 Summary

Our third sprint started on Monday, November 6th and featured 30 story points. In this week we sought to wrap up our major map interactions and introduce the infrastructure for our data visualizations. We were able to introduce a middle layer of our map, now grouping zip codes by their cities, then drilling into a city to expose the zip code data, and we were able to complete the map interactions that carried over from last sprint. In this sprint we also introduced a proof-of-concept functionality that featured tabs below the map which provided a place to post information relevant to the location the user was exploring. These tabs are dynamic and can change according to where the user is clicking into and what level of drilling, they currently have. Finally, we gained more experience understanding customer needs, and communicating plans, as we developed Figma mockups showcasing our plans for our tabbing feature moving forward. The only issue we were unable to complete was our GitHub setup again, it seems that although we have accounts, there is some approval issue that we do not have access to the repository.

Also, during this week, we visited the Fidelity offices in Boston and Rhode Island. This was a great experience being able to see the offices in person, meet some of our Fidelity contacts, and learn more about the company. We also got a chance to practice for our final presentation, as we held our weekly meeting in a conference room and used the conference call features that were available to us in the offices. Janette was able to grab her laptop from the Boston offices, and both Sandra and Janette were able to set up their laptops with the required python packages to run the code. The only technological issue that occurred was that Jack's virtual machine got tagged for a possible cyber security threat. However, this was not a significant issue as a temporary virtual machine was provided and it did not take long to set up the new machine. Overall, we were able to complete 30 of 31 assigned story points, with the GitHub setup continuing into the next sprint.

User Story	Subtasks	Points	Assignees	Status
Informative Tabs	- Create a tabbing feature to host relevant information/popups - Build out tabs to provide static information on the map - Connect dataset to the tabs and create a dynamic tab to prove that they can update in real time with the map and with necessary data	8	All	Done
Figma Mockups	- Create a mockup to showcase our plans for the product going forward, with the tabbing feature in mind	5	Sandra	Done
Group Zips by City	- Aggregate zips by their respective cities - Create a middle map layer for city groupings - Allow for a city to be clicked on and drilled into	5	Jack	Done
Click Through Drill Down	- Get zip data - Create variables to track selections and current map parameters	5	Jack	Done

	- Split a selected state into zips			
Zoom In Drill Down	- Switch connections to CSVs - Allow selection of a state and split that state into zip map	5	Jack	Done
Setup GitHub	- Gain access to GitHub repo once set up by fidelity	1	All	In progress
Census Data Above 18	- Clean census data to only include populations above 18	2	Janette	Done

Table 7 – Sprint 3 User Stories

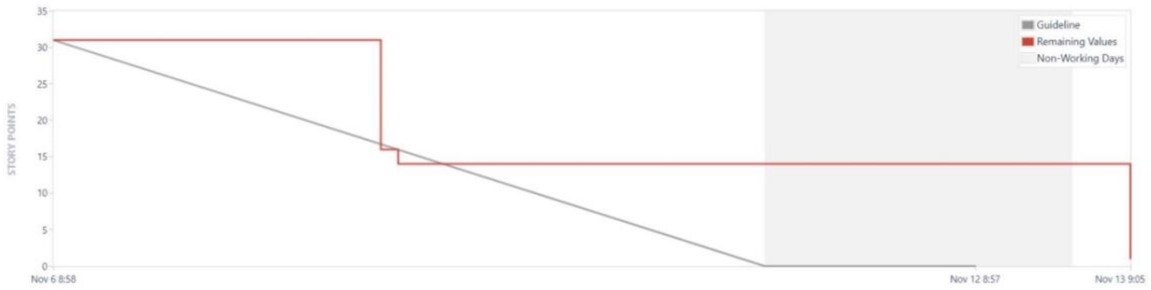

Figure 10 – Sprint 3 Burndown Chart

7.4.2 Retrospective

This week we were able to be very productive, however it should be restated that part of the reason we were able to complete so many story points is that 11 points carried over and had significant work on them from the last sprint. With that being said, our team started to reach its full potential for output this week as all laptops were set up. Up to this point we have been dedicated to our daily scrums and have maintained frequent communication within the team and with advisors. We believe that our quickness to reach out when blocked, or when needing guidance has helped our team maintain a good velocity, and we hope that in the following sprints.

We also noted that we need to keep our Jira updated, particularly since each team member now had the ability to run the source code and make changes. Adding on to this, our daily scrums at 1pm meant that team members would be working each day before we met, and we wanted to ensure that no team members would be working on the same task if an issue had not been moved to in progress.

7.5 Sprint 4: 11/13 - 11/20

7.5.1 Summary

Sprint 4 started on Monday, November 13th, and our focus for this sprint was to improve the user experience while also building out some of our informative tabs. We were able to complete our user experience improvements within our static map, as the dynamic maps still needed package fixes to implement styling features. We also made significant progress towards building out our informative tabs, however we did come across some blockers. The Fidelity BPM data that we were given did not explain the hierarchical structure of their columns, so when we made our initial pie chart and calculations, they included sections that did not belong together. However, this was discussed in our Fidelity sponsor meeting, and we were later given a hierarchy csv that we could use to roll up the data we were given. As a result, our tab stories were carried into the following sprint with the hope of completing them. Overall, we were able to complete 14 of 25 story points. The GitHub setup issue persisted through this week and at this point there wasn’t much that the team could do as it was an administrative issue, so in our retrospective we did not consider this story reflective of our process.

User Story	**Subtasks**	**Points**	**Assignees**	**Status**
Outline Shape on Hover	- Create a highlighting feature so that a user can clearly see the location they are hovering over	3	Jack	Done
Group Zips by County	- Obtain zip and county dataset for US - Aggregate zip data by their counties - Create a county map layer	3	Jack	Done
Check County GeoJson	- Investigate how many cases of overlapping zips there are	3	Jack	Done

	- Validate table merges and aggregation methods			
Change Color Mapping	- Adjust color scales to use colors not already used by map - Explore map tiles that use simplified colors	2	Jack	Done
Heatmapping in One Layer	- Using layering feature to include a checkbox of possible layers - Move heatmaps from individual maps into one and include the layer control	3	Jack	Done
Incorporate Region Demographics	- Build a dynamic tab to hold demographic info about current selected region - Explore python data visualization libraries - Implement visualizations to show region demographics	5	All	In Progress
Incorporate Business Line Info	- Build a dynamic tab to hold Fidelity business line info - Implement visualizations to show business line info respective to current selected region	5	All	In Progress
Setup GitHub	- Gain access to GitHub repo once set up by fidelity	1	All	In Progress

Table 8 – Sprint 4 User Stories

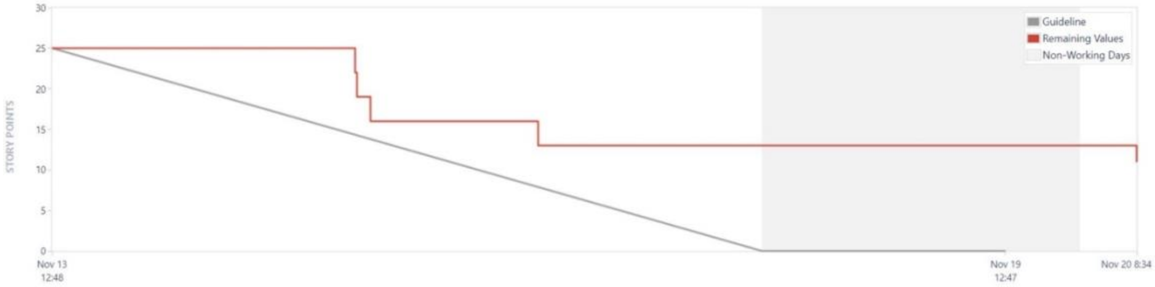

Figure 11 – Sprint 4 Burndown Chart

7.5.2 Retrospective

In this week's retrospective we noted that team members did a good job of updating Jira as tasked were being worked. This allowed the team to have up to date information on the state of the current sprint before our daily scrums. We also noted that although GitHub continued to be an issue, the nature of our tabbing feature allows members to write code separately and later incorporate that code into a main file, acting as our own repository managers. However, to do this, communication needed to be frequent and clear, which we had maintained at a high level up to this point.

Outside of the sprint stories, we also noted that communication with our Fidelity sponsors had been efficient. In the previous week's meeting, we voiced that we would be ready for the BPM data soon, and we were able to get those CSVs for this week, highlighting the importance of being ahead of our needs so that way the time zone difference cannot block us from completing our work.

For areas of improvement, we decided that we need to update our shared code more often, like we would if we had access to a GitHub repository. This will ensure that as changes are made, all members are updated, and their code isn't outdated once they finish working on an issue. We also agreed that we should start dedicating some user story points in following sprints to our MQP report as this is occasionally a second thought behind completing the current sprint. By dedicating story points to the report, we can ensure that our report stays on track, and it allows flexibility in the case of blockers that prevent us from improving our code.

7.6 Sprint 5: 11/20 - 11/27

7.6.1 Summary

Sprint 5 started on Monday, November 20th. Our focus this sprint was to wrap up the demographic visualizations that we had started last sprint, and to update this report. Since this sprint fell on a holiday, Thanksgiving, we did not include other story points as team members had limited time this week. This week we were provided with an updated dataset that included the Line of Business (LOB), and LOB grouping for each entry. This hierarchy aided us in revising our current LOB tab, as we were able to fix the hierarchy structure for our sunburst chart, and we were able to accurately aggregate asset amounts according to this hierarchy. Using this data, we also were able to update our demographic tab, which now featured a stacked bar chart of Fidelity's assets, with the x axis being split by age group, and the bars being split by gender. Lastly, we updated our report, according to our advisors' comments, and updates with the project. It should also be noted that the GitHub setup made some progress as we now had access to our accounts, but the team did not have access to the repository. Overall, we completed 15 of 16 total story points.

User Story	Subtasks	Points	Assignees	Status
Incorporate Region Demographics	- Process new CSV - Aggregate data based on age group, and gender - Load data into stacked bar chart	5	Jack	Done
Incorporate Business Line Info	- Connect to new CSV - Aggregate data based on LOB Group, and LOB - Implement sunburst chart and load data into it	5	Jack	Done
Revise MQP Report	- Update report according to project - Revise report based on advisor comments	5	All	Done
Setup GitHub	- Gain access to GitHub repo once set up by fidelity	1	All	In Progress

Table 9 – Sprint 5 User Stories

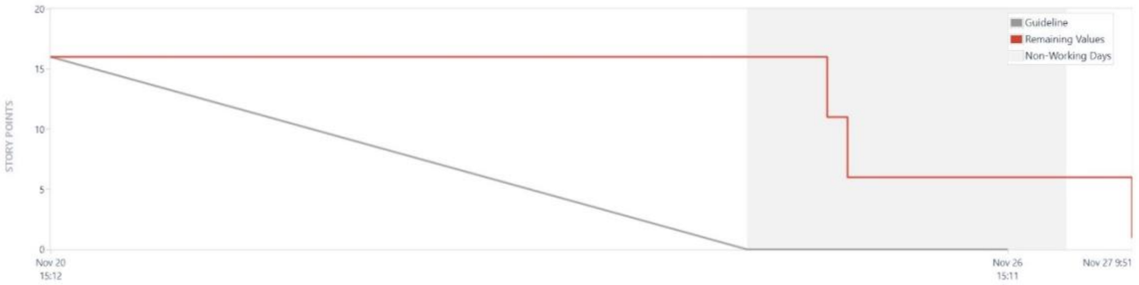

Figure 12 – Sprint 5 Burndown Chart

7.6.2 Retrospective

For this sprint the team felt that we did a good job of completing our tasks even with the shortened week due to the holiday. We also noted that communication remained at a high level. One thing that was pointed out to us by the advisors is that last sprint, even when we did not have the correct hierarchies for the LOBs we still guessed and made a diagram for it. Then in our meetings, those assumptions were corrected, and we were able to get feedback on the diagram itself. In doing so, the development this week was easier as we now had a better understanding of the LOB structure, and we had a working tab that we could fix. This highlighted the importance of proving/disproving assumptions, and how learning can be had even when there are roadblocks. We also felt that including a story for revising our report helped to keep us on track and decided that we will continue to include report stories going forward.

7.7 Sprint 6: 11/27 - 12/04

7.7.1 Summary

We started sprint 6 on Monday, November 27th, with the goal of making the last major improvements to the map this week, as at the end of the following week we would present our map to Fidelity. This included adding more button functionality to the map so that a user could navigate more seamlessly. We also fixed some bugs with the dynamic tabs as before, some of the data was not aggregating properly, or there were missing attributes. After fixing these tabs we then took time to refine the LOB and Demographic tabs. For the LOB tab we included filters for gender and age, allowing for more insight into LOB splits as a user is exploring the map. We also wrote out the total asset amounts for each LOB parent and their children, so that the dollar amounts could be seen more easily, rather than having to hover over each section of the chart.

For the demographic tabs we included a drop-down selection, which would let a user filter the stacked by chart by a LOB group, and from there each individual LOB would also be displayed. Lastly, the packages that we were using were updated, and styling could now be added to our dynamic map, so we took time to update our dependencies, and bring both an asset heatmap, and percent heatmap into our dynamic map. After making progress on our map, we took more time to update our report. Outside of the map we also took time in Power BI to explore and visualize the investor center data that was given to us. It should also be noted that we did not have access to the GitHub repository, however we had grown accustomed to sharing code, and as mentioned before the nature of the tabbing feature allowed updates to be made and shared without the need for GitHub, so we opted to continue developing how we have been, sharing updates between ourselves as we go. Overall, we completed all 28 story points this week.

User Story	Subtasks	Points	Assignees	Status
Investor Center Report	- Load data into power bi - Create visualizations - Explore more capabilities	5	Sandra	Done
Add More Reset Options	- Create more buttons for drilling into and out of different geographies	2	Jack	Done
Fix Dynamic Example Table	- Find why data is not aggregating properly - Adjust calculations - Fix missing columns	2	Jack	Done
Refine LOB Tab	- Add filtering options based on demographics - Include assets for each LOB group and LOB written out below chart	5	Jack	Done
Add Hierarchy Structure to Demographic Tab	- Bring LOB data into current dataset - Include drop down select for LOB Group - Filter data on selection and include expanders for each individual LOB	5	Jack	Done

Bring Heatmap into Dynamic Map	- Update packages - Include drop down select for percent and assets - Add heatmap to dynamic map	5	Jack	Done
Revise MQP Report	- Update report according to project - Revise report based on advisor comments	3	All	Done
Setup GitHub	- Gain access to repository	1	All	No Longer a Story

Table 10 – Sprint 6 User Stories

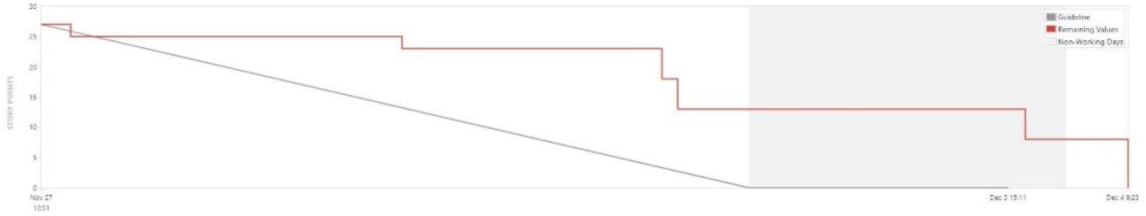

Figure 13 – Sprint 6 Burndown Chart

7.7.2 Retrospective

This week the team was able to complete a large amount of story points and did not have any stories carry over to the next sprint. We felt that this was one of our most productive weeks. Throughout this week we communicated with advisors both at WPI and at Fidelity early, in order to answer any questions we had and avoid hitting any roadblocks. We also felt that we did a good job of doing getting a lot of tasks in progress early. This meant that if there were issues or roadblocks, other stories could be completed while waiting to be unblocked. This helped us to keep the project on track and lead to us completing the entire sprint.

7.8 Sprint 7: 12/04 - 12/11

7.8.1 Summary

Sprint 7 started on Monday, December 4th, and had the goal of making our final revisions to the map before presenting it on Friday December 8th. For this week we focused on small adjustments for the map rather than large changes as it needed to be ready to present for a practice run Thursday morning, and a final demo on Friday. To start with we noticed that there were issues with drilling back up to a county level after viewing the map at the zip level. We also took time to create drop down selections for states and counties, so that way a user could search for a specific geography without needing to know where on a map that geographic location is. We also took time to switch all of our data connections to the most up to date CSVs provided to us so that way we would be displaying reliable information in our application. With this new data set there were millions of entries, and as a result loading and processing this data took some time while running the app. To speed this up, we decided to create pre-drilled CSVs for each state, as well as a pre-aggregated CSV so that way there was less wait time when navigating the map. We also included the investor center proximity data that was shared with us in our tabs. Lastly, we wanted to start a statistical report on the proximity data to show how machine learning could be applied to the dataset, and as a view into the next steps for this application. Overall, we were able to complete 18 of 23 story points, with the statistical report continuing into our final week.

User Story	Subtasks	Points	Assignees	Status
Add Drop Down Selects	- Include drop down selects for state and county - Bind these values to the app state variables	2	Jack	Done
Create Pre-Drilled CSVs	- Aggregate data at the national level for both demographics and LOBs - Split data by state, and create new CSVs for each state	2	Jack	Done
Include Investor Center Data	- Load proximity data into app - Include this in displayed datasets	2	Jack	Done
Switch All Data to New CSVs	- Switch all methods to connect to new CSVs	2	Jack	Done
Fix Drill Up Options	- Find where app state is persisting - Fix update methods	5	Jack	Done

Revise MQP Report	- Update report according to project - Revise report based on advisor comments	5	All	Done
Statistical Report	- Create new page for report - Explore dataset - Clean data and perform analyses - Report findings	5	All	In Progress

Table 11 – Sprint 7 User Stories

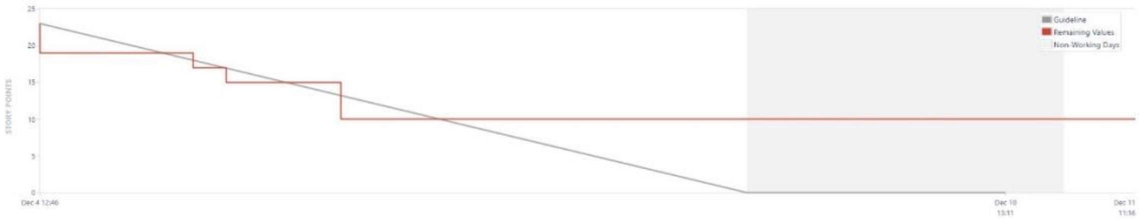

Figure 14 – Sprint 7 Burndown Chart

7.8.2 Retrospective

One of the challenges heading into this week was to make progress on the application, without having too much to do that we would not finish before the presentation. We felt that we created a good balance of still making progress on this shortened week, while also finishing work on the map with enough time to prepare for our presentation. We were unable to finish the statistical report, however, we talked with our Fidelity sponsors, and we were able to extend this report one more week, as it did not affect the map, and it was a good way to showcase future work.

7.9 Sprint 8: 12/11 - 12/13

7.9.1 Summary

This sprint was our 8th and final sprint. It started on Monday December 11th, and we aimed to wrap up the project and the paper. For the project we finished up our statistical report, which included performing linear and polynomial regressions on the relationship between a zip codes distance to the closest investor center, and the percentage of customers to population within that zip code. We then shared this report with our Fidelity sponsors and focused on wrapping up our report. Lastly, we added a story for the Power BI dashboard that we also worked on last week but forgot to include it from the back log. We completed 18 story points this week and completed our project.

User Story	Subtasks	Points	Assignees	Status
Statistical Report	- Perform regressions on scatter plots - Analyze results	5	Jack	Done
Power BI Dashboard	- Finalize dashboard	5	Sandra	Done
Revise MQP Report	- Wrap up report - Get last revisions from advisors - Get it approved by Fidelity Sponsors	8	All	Done

Table 12 – Sprint 8 User Stories

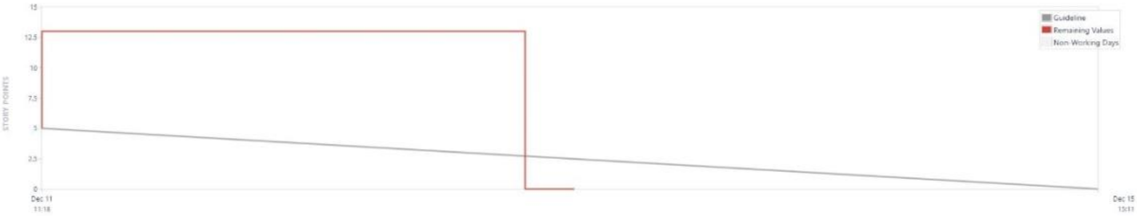

Figure 15 – Sprint 8 Burndown Chart

7.9.2 Retrospective

We were able to complete all tasks this week and wrap up our project. Looking back at the overall project we felt that we were able to maintain a high level of communication throughout each sprint. We also felt that we were able to adapt and change our ways to keep productivity high, like including report stories in the later sprints, and adjusting and planning for the time difference between us and the India team. By being flexible and communicative, we were able to become a well-functioning team and consistently produce value each week.

7.10 Velocity Chart and Cumulative Flow Diagram

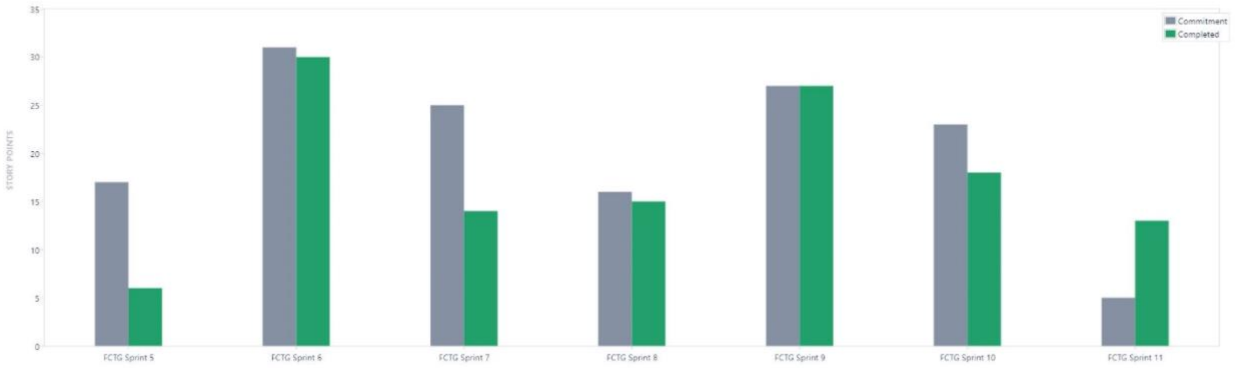

Figure 16 – Project Velocity Chart

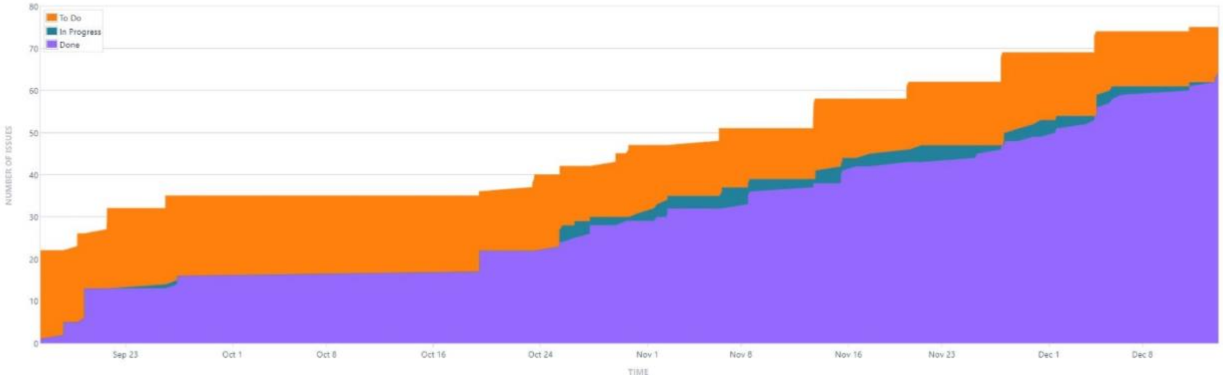

Figure 17 – Cumulative Flow Diagram

7.10.1 Summary

Shown above is our team's velocity chart and cumulative flow diagram over the course of the project. Since our Jira included PQP our first sprint of MQP is labeled as "Sprint 4". It should be noted that the velocity chart only includes the 7 most recent sprints, so the first sprint of the MQP, "Sprint 4", is not included. It should also be noted that since the cumulative flow diagram includes our PQP there is a flat portion of the graph until we started our project in late October. From these charts we can see that there were some weeks where we had story points carrying over, and we can also see that some weeks had more story points completed than others. This indicates that our team could have done better to accurately score different user stories, which in turn could have helped us understand our point capacity for each sprint better. One interesting finding is that after the 4th sprint, "Sprint 7", where we had 11 points carry over, we did not have another week with as large of a carry over. The next highest was in our 2nd to last sprint which only had one story of 5 points carry over. This could indicate that after that point we started focusing on completing the larger tasks earlier in each sprint.

8. Assessment

8.1 Jack’s Assessment

Looking back at the total project, I see it as a success. Looking just at our product, we were able to develop a powerful tool for exploring and analyzing Fidelity’s Business Performance Measurement data. A major focus of this project was the geo-spatial aspect to the dataset. Our application allows a user to explore geographies across the United States at various grain levels and provides informative tabs relevant to the geography that user is exploring. While we did not have time to introduce time-series analyses for customer acquisition, or customer profitability, we still provided extensive functionalities. In our tabbing feature a user can explore the aggregated data for that location, tabulated data of the next grain level, and data visualizations featuring Line-of-Business hierarchies, and customer demographics. We also showcased how the application could be expanded to introduce AI and machine learning through our regression analyses. This application can act as an example for future applications centered around other datasets and can also act as a framework for incorporating future improvements.

Outside of product goals, our project also provided valuable experiences as a student. Through other projects I have been previously exposed to Agile development, serving as a scrum master. However, this project gave me a new perspective serving as the product owner which allowed me to deepen my understanding of Agile, and gain more familiarity with Jira, a powerful project management tool. Through the development process I have also sharpened my skills as a team member. Working in a three-person team, communication and organization must be high. Along with this, there were times when one person on the team was very capable of a certain task that other members were not familiar with. This led to practice both teaching others and learning from others. Finally, I have always praised project-based learning. I see this type of learning as a way to apply what you have learned in a classroom, to something tangible that exists in the real world. Often times this can show you where you might need to spend more time studying and practicing, while also providing you a chance to improve a variety of skills. I found this project to be a great experience to grow as a developer and data scientist, and it has given me a newfound passion for the intersection between finance, software engineering, and data science.

8.2 Sandra’s Assessment

Reviewing the project, it's clear we as a team have accomplished something meaningful. Our team successfully developed a strong tool for exploring Fidelity's Business Performance Measurement (BPM) data, with a particular emphasis on the geo-spatial aspect. Users now have the ability to navigate U.S. geographies through personalized tabs, offering insights specific to each location. While we didn't manage to squeeze in time-series analyses for customer acquisition or profitability, our application still delivers inclusive functionalities. Tabs provide aggregated data, tabulated information for different granularity levels, and data visualizations featuring Line-of-Business (LOB) hierarchies and customer demographics. Additionally, we demonstrated the potential integration of AI and machine learning through regression analyses, positioning our application as a practical example for future projects and a framework for continuous improvement.

Beyond the technical achievements, this project has been a valuable learning experience. Seeing the views from a scrum master and product owner provided deep insights into Agile practices and working with Jira for project management collaboration understanding. Within our three-person team, effective communication and leveraging individual strengths were crucial. This collaborative approach helped mutual learning and skill refinement. The project's commitment to a project-based learning approach, translating academic knowledge into practical applications, has been worthwhile and has contributed significantly to personal and professional growth.

Post-project completion, it's evident our team met its objectives. The primary goal of presenting customer demographics and market shares through an interactive geospatial map was accomplished. The progression from basic state breakdowns to a detailed BPM map with drill-down capabilities showcased the team's adaptability and commitment to delivering a high-quality solution. On a personal note, the project expanded my skill set, introducing me to Streamlit, Folium, PowerBI, and emphasized the importance of collaborative teamwork in delivering comprehensive solutions aligned with project requirements.

8.3 Janette's Assessment

After the completion of our project, I believe that we were successful meeting our goals. Our primary goal was to communicate customer demographics and market shares in the form of a geospatial map. This map was aimed to be interactive for and easy for the user to manipulate to access information as quickly as possible. By using Streamlit & Folium, we were able to generate a map with the necessary information based on the original prototypes; given to us already were the States, Zips and States Zips maps. These served as basic initial steppingstones for us to work off of. The first map we created was the Asset map, which was very basic and self-explanatory - it simply broke down asset amounts by state. However, the issue with this map was that every time we created a new state, the entire map would re-render and it took quite a while for the data to sync up. We realized that this issue in combination with operating the system from a virtual machine was very difficult. This led to us building a Layer map - which combines both the asset and percentage maps into one. This map was the first one that provided some drill down functionality, so we got a chance to explore some aspects of interaction with the map with city-level grouping. using a city drill down.

However, this initiative was not the most ideal, seeing as many zip codes belonged to the same city. For example, Worcester contains various zips with locations that are mere streets apart. this is why we switched over to county grouping as we felt it would be more representative of what we wanted to accomplish. That led to us creating our BPM map, where we drilled down by county and zip and worked on providing more attributes so the user can filter by either percentage or asset. This map was where we did the most work. We were able to draw valuable conclusions from the map, ultimately allowing us to better understand Fidelity as a company and what customers need from them. I was able to retain a lot of important skills from this project. Streamlit & Folium were completely new topics for me, and I had never had the opportunity to explore the capabilities of this interface. While I am familiar with the Tableau map functionality, I had never understood how to generate a map through Python from the ground-up, and this project allowed me to do so. Additionally, I was able to work closely with Sandra & Jack through the past 7-weeks, teaching me the importance of leveraging everyone's skills and weaknesses to create a deliverable that encapsulated the requirements as best we could.

9. Business and Risk Management

Should the project be successful for the company, they will have the ability to drill-down their customer information to understand specific details about demographics they would like to focus on or expand to. Additionally, they will not have a strong understanding of the demographics that the majority of their profits come from. By having the option to further analyze customer information, the company can understand which demographics they need to appeal more to in order to maximize revenue. This may include expanding geographically or altering marketing techniques in order to best suit the needs of the target demographic. As it stands, this drill-down method can be instituted in future years so that the company can analyze which customers prove to be more consistent, as well as have a visual option to determine trends and patterns in customer behavior.

In the preliminary stages of executing our project, we faced difficulty accessing Fidelity data systems due to the risk that is associated with pulling company data on personal computers. For example, we were unable to use the Snowflake database to select datasets and were instead provided individual datasets that we could manipulate. Additionally, we were asked to use a two-factor authenticator specific to the company, as well as a virtual desktop. While the usage of a virtual desktop was convenient for us to access the required applications remotely, we found that using a remote desktop significantly decreased the speed of certain applications and code compilation, which we found to be an obstacle at times. Alternatively, by obtaining laptops for two of the team members, this issue was able to be rectified.

Working within the business allowed the team to be exposed to various types of risks. In terms of operational risks, we realized that there were various cybersecurity measures that we had to be mindful of – including accessing company data, using Fidelity laptops, and issues faced with the virtual machine. While we were not directly working with the Finance team, we learned the major ways in which they market their metrics to appeal to the customer, as well as the flow of sales that exists within the company. With every corporate office exists some aspect of a physical repetition risk, but seeing as we were only involved with the project for a short 7-weeks, we did not feel the impacts of such risks that our co-parts at Fidelity experienced.

Having been under the leadership of various team members, we understood the value of change management risks and how to overcome them by catering to the needs and expectations of each team member in a unique way. We were able to identify the different styles of leadership and note which team member to go to with specific inquiries. In terms of risk involved with training, we did not undergo any formal training or onboarding to the company but viewed walkthrough examples of our project goals and objectives and had soft training from our coworkers. Additionally, the team was able to teach one another about various skills that each one of them possessed; for example, some of us were stronger in Software Engineering, others with presentation and curating mock-ups, etc.

By completing our MQP, our team was able to witness corporate culture and employee life at Fidelity in both the Boston & Rhode Island offices. We were exposed to the schedule of Fidelity employees, while equally learning the importance of staying organized while performing remote work. In this sense, we were able to explore our team dynamics and shift our work schedule and the style in which we conducted our Sprint meetings. Throughout the project, we were exposed to various approaches to problems from Computer Science, Data Science and Business perspectives. In terms of action items that can be taken to improve this project in the near future, we found that technological barriers were a large part of the delays in our Sprint progress and overall business model. Should this project be continued or replicated by other students, we would recommend for all team members to obtain Fidelity issued laptops prior to the project’s start date – preferably in the onboarding period when background checks occurred – to ensure that everyone is utilizing the same hardware throughout the project. By having a standardized medium of technology, this not only eliminates the need for a virtual machine, but also allows everyone to have access to the same software packages and applications and go through the same process to download additional software as well.

10. Management Information Systems Major Requirements

10.1 Entity Relationship Diagram (ERD)

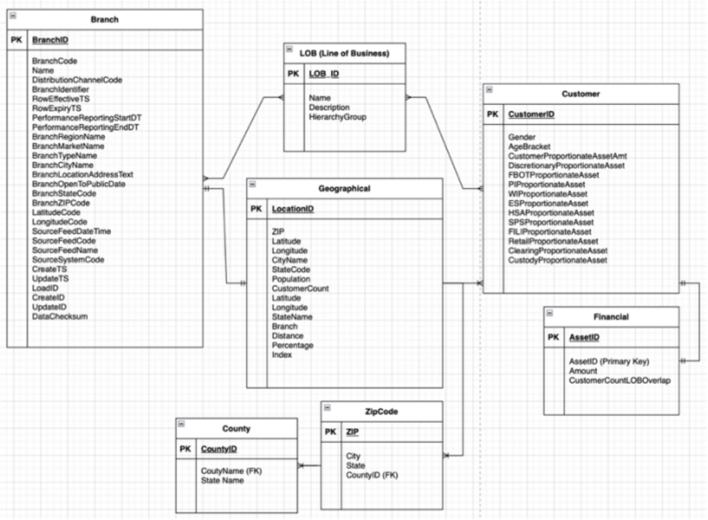

Figure 18 – ERD

In the domain of data management and database design, Entity-Relationship Diagrams (ERDs) serve as critical tools for comprehending and structuring the interconnections among different elements. These diagrams provide a viewpoint on relationships within a system, ensuring that the data organization aligns exactly with the operational requirements of a business.

Entities serve as the foundation of an ERD, acting as the essential components in a database model. They encapsulate both concrete entities like customers or products and conceptual elements like addresses. Each entity has unique characteristics known as attributes, providing specific details about the entity. For instance, in the shown ERD, the 'Customer' entity contains attributes such as 'CustomerID,' 'Gender,' and 'AgeBracket.' These attributes define and characterize the entity, forming the basis for a structured and organized data model. Attributes serve as specific details about entities, offering a closer look at each element. Examples in our databases include details like 'CustomerProportionateAssetAmt' or 'DiscretionaryProportionateAsset,' providing additional information about individual entities.

Primary keys (PK) and foreign keys (FK) play a pivotal role in ERDs. Primary keys are unique identifiers for each record in an entity, making sure that no two entities share identical details. On the other side, foreign keys create links between entities, letting one entity refer to another. In our database, 'CustomerID' might be the primary key for the 'Customer' entity, while 'ZIP' could be a foreign key connecting 'Location' to 'Customer.'

The relationships in ERDs show how entities are connected, outlining how information flows between them. For example, a 'Customer' entity might be linked to a 'Location' entity using a ZIP code, telling us where each customer is located. When examining the entities, there is a network of relationships shown. For instance, a 'Customer' entity could be directly linked to an 'Asset' entity in a one-to-one connection, documenting the specific financial details associated with each customer. Together, the 'Customer' entity might engage in a many-to-many relationship with a 'Line of Business' entity, illustrating the varied financial services utilized by customers.

Branches play a significant role by forming connections with geographical entities. The 'Branch' entity, for instance, could establish a one-to-one relationship with a 'Geographical' entity, offering accurate coordinates for each branch. This relationship opens up possibilities for location-based analyses, similar to the map we generated earlier.

Geographical entities bring a wider perspective to the story. A ZIP code in the 'Location' entity might have a many-to-one link with a 'County,' showing us the bigger geographical picture. Similarly, 'Location' might connect with a 'Customer' entity through an interaction of many-to-many relationships.

In the bigger picture of ERDs, every entity, attribute, and relationship comes together to create a smooth flow of information. The 'Customer' entity, packed with financial details, connects with the 'Line of Business’ entity, giving us a glimpse into the various financial services customers use. At the same time, the 'Branch' entity tells its story with geographical entities, giving us more location-based insights.

10.2 Data Flow Diagram (DFD):

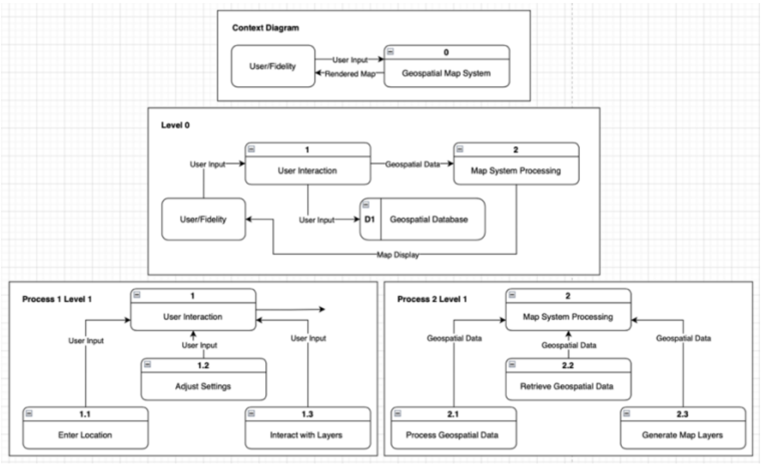

Figure 19 - DFD

10.2.1 Context Diagram

The context diagram serves as an insightful representation of our geospatial map system and its interactions with external entities, with a primary focus on the end-users. Visualizing the system as a 'black box' allows us to emphasize the exchanges between the system and its external environment while restructuring the complex internal processes.

The external entity, named 'User/Fidelity,' stands for the people using the geospatial map system. In this interaction, the Geospatial Map System plays a pivotal role as it highlights the inner workings to emphasize system limits and key functions. This diagram serves as a helpful tool to easily understand the important components and their interconnections.

The "User Input" data flow provides a pathway for information from the User to the Geospatial Map System. This flow signifies the inputs provided by users, including their preferences and specified locations. It highlights the interactive features of the system, where user interactions influence how the system operates. Equally, the "Rendered Map" data flow illustrates the reverse journey, from the Geospatial Map System back to the User. This flow delivers the end product – the visualized map – to the users, completing the interaction loop.

This context diagram, while maintaining simplicity, offers a detailed view of how user inputs prompt actions within the system, resulting in a definite output for the users. It lays the foundation for further exploration by providing a clear picture of the external interactions and the vital role users play in navigating the geospatial map system.

10.2.2 Level 0

At Level 0 of the Data Flow Diagram (DFD), we explore into the foundational processes that drive the interaction and data flow within our geospatial map system. Two key processes that take place are 'User Interaction' and 'Map System Processing' which directing the exchange of information.

In the 'User Interaction' process, users actively engage with the system, providing input through the 'User Input' data flow. This input includes user preferences, specifications, and location details, initiating an interaction between the user and the system. The 'Map System Processing' process takes charge of handling this input, using geospatial data in accordance with system requirements.

The 'Geospatial Data' data flow acts as a bridge between 'User Interaction' and 'Map System Processing,' carrying relevant information to be processed. As the system processes this data, it generates the 'Map Display' data flow, delivering the visualized map output to the user. This recurring flow ensures a responsive user experience.

In addition to these processes, the 'Geospatial Database' serves as the data store for geospatial information. Both 'User Interaction' and 'Map System Processing' contribute to and draw from this central repository contributing and retrieving files, therefore ensuring reliable data management.

In summary, the Level 0 DFD explains the core processes, data flows, and data stores that enable a dynamic interaction between users and the geospatial map system. It establishes the footing for further exploration, offering a detailed overview of the relationships within the system architecture.

10.2.3 Level 1: User Interaction

At the next level of our map system breakdown, 'User Interaction' emerges as the pivotal process shaping how users engage with our geospatial map. It functions as the central hub, coordinating three main actions: 'Enter Location,' 'Adjust Settings,' and 'Interact with Layers.'

To elaborate, 'Enter Location' becomes a portal where users not only pinpoint spots on the map but also utilize a search bar to look up specific states or counties. This feature enhances user convenience, allowing for a more fitted experience. Next, 'Adjust Settings' lets users to customize functions like map details, ensuring they can easily spot what they're looking for. This personalized touch enhances the user's ability to make the map to their needs. Finally, 'Interact with Layers' takes users through different heat map layers, providing a layered and detailed experience.

Moving from 'User Interaction' to 'Map System Processing' involves a key data flow – the 'User Input.' This flow carries user actions into the system, navigating the map's updates. This setup guarantees the map responds promptly to every user move, fostering an interactive experience that aligns with the expectations of the users.

10.2.4 Level 1: Map System Processing

The Level 1 DFDs provide a granular breakdown, illuminating the specific actions, involved data movements, and the web of relationships within 'Map System Processing.' Our focus increases on the importance of ‘'Map System Processing,’ as we look into Level 1 in DFD unfolding with the complex flow of subprocesses and data flows that underpin its functionality. ‘Map system processing’ carries the responsibility for the accurate and efficient use of geospatial data, ensuring a seamless and efficient user experience.

The subprocesses within this level, specifically 'Retrieve Geospatial Data,' 'Process Geospatial Data,' and 'Generate Map Layers,' all together contribute to the detailed functioning of the entire system. 'Retrieve Geospatial Data' plays an important role, actively captivating relevant information based on the user's specified location. This marks a faster flow of data into the core processing mechanism, setting the stage for the next level of refined performance.

After this, the attention shifts to 'Process Geospatial Data,' where tasks unfold to ensure that geospatial information undergoes filtering, aggregation, or transformation, aligning with the system's needs. The 'Generate Map Layers' sub-process brings variety by crafting various map layers based on the processed geospatial data. The data flows within these sub-processes are directed toward the goal — 'Map Display.' This crucial step ensures that users are presented with visually rendered map layers, creating an interactive environment.

Furthermore, the symbiotic relationship between 'Map System Processing' and 'Data Management' is established through the 'Geospatial Data' data flow. This connection to the data store adds depth to the system's functionality, contributing to its comprehensive and dynamic character. The system's ability to interact with and draw from the data store enhances reliability and responsiveness.

11. Results

11.1 Data Exploration and Cleaning

After completing the development of the application, time was dedicated to applying regression analyses to some of the dataset. Our sponsors indicated that they were specifically interested in discovering any insights into the relationship between geographical proximity to an investor center, and the total assets, customers, and percentage of customers to the population for that geographical area. Our sponsors provided us with proximity data for each zip code that had Fidelity customers associated with it. We then aggregated the BPM data by zip code and joined these datasets together. This combined dataset which included total customers, total assets, population, percentage of customers to population, proximity to closest investor center, and state, for each zip code, was then analyzed.

To explore these relationships the first thing we did was create three scatter plots with each independent variable being the proximity attribute, and the response variables being total customers, percentage, and total assets. From these initial scatter plots there seemed to be somewhat strong negative relationships, meaning that as a zip code was further from an investor center, the less customers, assets, and lower percentage that zip code tended to have. However, we did notice that there were a few outliers in the percentage scatter plot, with some percentages being higher than 100. We then decided to investigate these irregularities.

We created a new dataset of every entry whose percentage was higher than 100 and found that for each zip code, the population totals were very low. We determined that these population totals could be low as the data came from the most recent US census and it is possible that some geographies did not receive enough coverage to provide accurate data. As a result, these low population totals would in turn provide inaccurate percentage calculations. Going from this we cleaned the data removing any zip code whose percentage was above 100, and we also removed any zip code whose population total was lower than 500 as a precaution to help avoid further inaccurate percentage calculations. The new scatter plot with the cleaned dataset showed a much more clear, negative, relationship between proximity and percentage.

11.2 Applying Regressions

After cleaning the dataset, we then decided to develop regressions on the proximity and percentage relationship and see how accurately we could predict a zip code's percentage based on its proximity to the closest investor center. It should be noted that we opted to not explore the relationship between total assets and proximity further as we felt that this relationship could be affected by a variety of other factors like median incomes of a zip code, and the size of a zip code. Similarly, we chose not to explore the relationship between total customers and proximity further as again the zip code size could affect the total customers, and the percentage attribute encapsulates the total customers in its calculation.

We first developed a simple linear regression using a train and test data split of 75% and 25%, respectively. The R-squared for the fitted regression on the test data came back very low, indicating a weak ability to predict the percentage of a zip code given the proximity to the closest investor center using a simple linear model. Looking back at the scatter plot we did notice that the relationship seemed to be more non-linear, as at the very close proximity zip codes the percentages grew very fast, and as zip codes approached the largest distances, their percentages seemed to plateau.

Considering this we then decided to train a polynomial regression of degree 3. We again used proximity as our only independent variable and percentage as the response, with a train and test split of 75% and 25%. The R-squared for the test data on the fitted regression did slightly better, however this difference could be attributed to the randomness in the experiment. We did notice that the quadratic coefficient was the largest, and as a result there was a slight curvature to the regression, but it did remain very close to the linear estimate.

For our final calculations we decided to consider a zip code’s state, as some states may have less dense populations and larger zip codes by area, meaning longer distances are more regular, and vice versa. To do this we iterated through each state that occurred in our cleaned data set. On each iteration we created a new dataset that took only zip codes in the current state from the cleaned data set. We then performed a linear regression with all that state’s zip codes, using the same training and test splits, and then stored the R-squared test scores for each state. We also only considered states who had at least 100 zip codes to avoid too small samples. This method effectively acted as a multi-linear regression considering state as another independent variable with proximity, without having to create 50 dummy variables to calculate the linear regression. We found that many states had higher R-squared values than the original linear regression, however only a few states were vastly higher.

12. Future Work

12.1 Application Speed

An important area of improvement for our product is the application's speed. While drilling up and down the map, and exploring different regions there is a large amount of data that needs to be loaded onto the web page. As a result, there can be some wait times as the web page loads the map, calculations, and visualizations. Part of the issue is that our connections to the data sets were through large CSV files, which can take lots of computation time to open, read, and store as a DataFrame. We were able to remedy this partially by aggregating and segregating datasets based on state, allowing for smaller CSVs to be loaded depending on where on the map a user was and what level of drilling that user was currently at. However, this could still be improved by connecting the webpage to a database like snowflake. This would allow specific queries to be made, so only needed data would be read at the immediate moment it is needed. Going along with this, these requests could be multi-threaded allowing for the application to load different contents as their requests are filled, rather than waiting for the entire page to be rendered.

12.2 Expanding Data

Another area of improvement is to expand the data set globally. As it stands, our work covers demographic data of the United States. After connecting with the offshore team in India, we realized it would be valuable if this work could extend to other parts of the world as well. If that is properly executed, there would be opportunities to compare market shares and business performance on a country or continent level, and see which geographies need to be focused on outside of just the US. The inclusion of global data would still work with our current packages as the Streamlit-Folium maps include the rest of the world. However, an important consideration is that the current drill levels are created with US geographies in mind (counties, and zip codes), and there may not be geographical equivalents in other countries around the world. So, there may have to be specific geographical groupings depending on the country a user is exploring, which may require specific geographic knowledge of different areas.

The data could also be expanded to include more external datasets. For our application we included two datasets external to the BPM dataset, those being the US ACS census data, and investor center proximity data. By including these datasets, we added valuable insights, but also proved that the code is portable enough to include any external datasets if they can be attributed to a state, county, or zip code. By including more datasets, the users of this system could be able to generate even more insights.

12.3 Improving Regressions

As stated in the results section our various regressions on zip code proximity to an investor center and a zip code's BPM data were able to indicate some correlations, but we were unable to develop strong predictions. We were able to get stronger prediction scores when we considered a zip code's state, effectively acting as a multi linear regression using proximity and state as independent variables. This identified another area of future work which would be to extend our regressions analyses into multi-linear regressions that considered more independent variables. Some of these variables could include a zip code's median income, as wealthier areas might tend to have more assets with fidelity, or possibly considering competitor investor center proximities, as more competition could affect market share in different zip codes. By including more explanatory variables, the relationships between them and the assets, customers, and percentages in different areas could be better understood, leading to more guided business decisions.

12.4 Expanding AI/ML

We introduced regressions to provide some analyses, however the application could also be expanded to include other AI or machine learning methods to provide valuables insights. Our applications page and tabbing structure allows the application to be organized, and for new sections to be added easily. Also, Streamlit being a python library allows developers to access python's powerful AI and ML libraries. For example, future developers could implement sentiment analysis across investor center reviews, social media, and other sources to mark geographies with high and low customer satisfaction. This could uncover trends or relationships that can influence the company’s performance in different areas. Another addition could include time series analyses for new customers across different geographies.

13. Conclusion

During our time at Fidelity, we were able to visit the Rhode Island and Boston offices and made many connections within the industry. This project allowed us to gain valuable insight into how Fidelity operates as a business and how various disciplines of study can converge under one business. Our goal was to explore the Business Performance Measurement data, populated with various customer-oriented attributes, and provide a way to discover valuable insights into this data. While we were provided with a map visual to work with, our goal was to improve it to include assets and customer count at the state, city, and zip levels. This was done using a Geo Spatial Map framework in combination with Python (Streamlit + Folium). With our updates, we were able to introduce a level of interactivity that would allow the user to use a drill-down method to view data from a state, county, or zip code perspective. We also incorporated other datasets into our application namely, the ACS Census dataset, and the Investor Center Proximity dataset. Through these datasets we were able to enhance the possible insights that this application can provide. We took time to develop tabs within our application that featured relevant tabulated data, Line-Of-Business splits, and customer demographics for a selected geography. Additionally, we provided a sample of what we might expect some future work on this application to look like through our application of regression analyses to help understand the relationship between different BPM variables.

Our development provides the ability to uncover valuable insights utilizing Fidelity’s Business Performance Measurement data. This project also provides a structure that allows future development and improvements to be implemented efficiently. Our team sees this product as a success, and while we did not reach all of the goals in our outline, we were able to lay the groundwork for that path going forward.

Along the way we had various technical difficulties in switching over to Fidelity-issued laptops, running code, and figuring out how to effectively present our progress in the form of mock-ups. There were also challenges when developing the code, aggregating datasets, and creating value for the user. However, this gave us great opportunities to learn and grow. Through this experience we became more knowledgeable of Fidelity and the broader FinTech space, we refined our coding abilities, improved our understandings of Agile development, and gained exposure to powerful technologies.

References

Atlassian. (n.d.-a). Welcome to Jira Software. Atlassian. https://www.atlassian.com/software/jira/guides/getting-started/introduction#what-is-jira-software

Atlassian. (n.d.-b). What is Agile?. Atlassian. https://www.atlassian.com/agile

Atlassian. (n.d.-c). What is Scrum?. Atlassian. https://www.atlassian.com/agile/scrum

Data Flow Diagram (DFD). Let's think critically about... (2011, August 20). https://trungtien.wordpress.com/2011/08/20/data-flow-diagram-dfd/

Data Flow Diagrams (DFDS) - black box concept. BrainMass. (n.d.). https://brainmass.com/computer-science/logic-design/data-flow-diagrams-dfds-black-box-concept-70145

Fidelity. (2023). Quarterly Updates Q3 2023 - Fidelity. Fidelity Investments. https://www.fidelity.com/about-fidelity/our-company/quarterly-updates/quarterlyupdates-q3-2023

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

Gallo, A. (2015, November 4). A Refresher on Regression Analysis. Harvard Business Review. https://hbr.org/2015/11/a-refresher-on-regression-analysis

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning: With Applications in R. Springer.

Microsoft. (n.d.). Get Started with Visual Studio Code. Visual Studio Code. https://code.visualstudio.com/learn

Python Software Foundation. (n.d.). About Python. Python.org. https://www.python.org/about/

Snowflake Inc. (n.d.). Why Snowflake Data Cloud. The Snowflake Data Cloud - Mobilize Data, Apps, and AI. https://www.snowflake.com/en/why-snowflake/

Streamlit Inc. (n.d.). Streamlit Documentation. https://docs.streamlit.io/

What is entity relationship diagram (ERD)? (n.d.). https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/