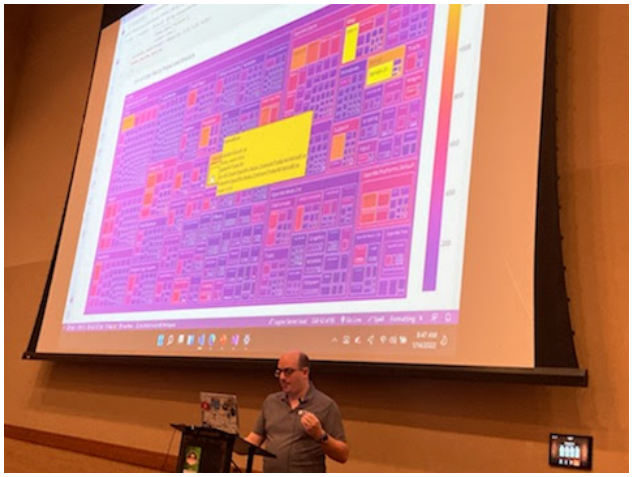
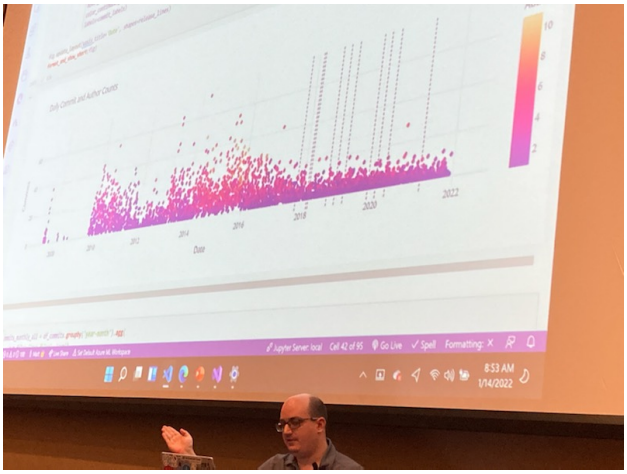
Reflection Paper

Personal impact

First of all, I want to cover what being able to pick my own project, my own technologies, and my own challenge level meant to me personally.

A week before my first classes at Franklin started I gave a talk at CodeMash 2022 where I showed the community an early look at my “GitStractor” tool that could extract CSV data from git repositories and visualize it in a git repository.





In a way, this talk served as the marker of the beginning of my studies,and it’s truly fitting for me to conclude my studies at Franklin by adding machine learning to a more refined version of this tool.

Additionally, during my time at Franklin I’ve grown from an eager learner to someone who teaches these things in the community enough to get official recognition from Microsoft and local community organizations and then again grown to the point where I’m now helping innovate the machine learning ecosystem for .NET developers using ML.NET and Polyglot Notebooks by authoring libraries on the topic as well as writing a book and creating a course, both releasing this fall.

But let’s zoom in a bit to just DATA-695 and discuss some of that journey.

Implementing analytics to solve a business problem

It’s not quite fair to say that I implemented analytics to solve a business problem here. GitStractor was an existing analytics solution that I created to solve a business problem I identified years ago: when you first join a new project, it can be hard to understand how that project has evolved, where it’s actively growing, what areas are stale, and who is involved in each major area.

GitStractor was built to solve that need and it did it quite well, but as I left teaching to become a consultant, I realized there were some gaps there in how well it could identify quality hotspots.

I knew from my own experience as a developer that commits were typically new features, bugfixes, or minor adjustments and tweaks, commonly called “chores”. The most interesting aspect of commits was the presence of bugs and what files needed to modify when bugs were fixed, so I

theorized that I could add a fast classification model to the tool to gain this additional information, but it would need to have a suitable labelled dataset for training.

I wasn't quite sure where to get the dataset until I started experimenting more with RAG and AI orchestration in semantic kernel over the winter. This gave me the idea that an LLM with the right prompt could generate candidate labels that could then be reviewed and adjusted as needed, reducing the time needed to set the initial labels. This process generally worked well and gave me the data I needed to begin my experiments.

The EDA process was quite valuable in validating my assumptions that correlations existed between different attributes of a commit and its ultimate classification.

This process also helped me explore ngram extraction both in word and character form. This was a new endeavor for me and so it was helpful to explore the results of extraction in the analysis phase before incorporating it into the machine learning phase.

Speaking of machine learning, during the first few weeks of the course I was working on part 2 of my upcoming book and really getting deep into ML.NET's internal pipelines and how to set up complex machine learning pipelines. Being able to work with ML.NET on this project was invaluable because I was able to push my knowledge further, and it turned out that being able to customize the model training pipeline was quite necessary for this project.

In particular, I needed to create a custom pipeline step to sanitize numbers and URLs in commit messages, and I needed to find a way of tweaking the default text processing pipeline to remove stop words. This required understanding the things that Microsoft was doing under the hood, which in turn required building a custom pipeline visualizer for Polyglot Notebooks.

I feel the project pushed my knowledge to new heights, let me develop new skills in myself, and new techniques for the community in the form of new libraries that are already publicly shared.

I also am elated to have already gotten some useful insights from the final model by integrating it into GitStractor and designing a few new visualizations to take advantage of the classification and probability.

Lessons learned

There are a few things I did on this project that I'd probably do differently.

First, my Phi-3 instruct 4k classifier I used to generate my initial candidate labels had a precision of 0.54 or thereabouts. This led to a large number of false positives in the training data due to the model being "too optimistic" in categorizing commits as bugfix commits.

I believe that tweaking my few-shot inferencing examples or the model's main instructional prompt could have reduced the rate of type I errors.

Secondly, it didn't take as long as I expected to review the commit labels and I should have expanded my training sample from 500 to 1000 or more rows.

Third, I think 5 source repositories were too few to include for training purposes. I don't think there was enough diversity in commit messages as a result, because the PFIs in some cases seemed biased towards particular parts of these repositories that were frequently broken. Were I to do this again, I'd likely select a broader set of repositories or ensure all repositories had roughly the same number of commits.

I was also somewhat surprised that some of the more advanced classification model trainers failed to eclipse the basic random forest and logistic regression models. Some of this could be training time, volume of data, or hyperparameter tuning. It's interesting to me that I generated over 60 pages of notebook results only to have my final model selection be a toss up between logistic regression and a random forest of all things!

Thoughts on DATA-695

As I mentioned in the open, I'm very thankful that this course afforded me the freedom to explore a topic that interested me and that I was allowed to work with advanced technologies of custom interest to myself.

When I look at Franklin and where I grew the most, it was where I was tackling assignments in ways that pushed my abilities as far as I could, whether that was reading a technical book a week on database internals in the database course, visualizing public repositories with GitStractor in the data visualization course, or building an image classification model to play rock paper scissors with me in the intro to Python course, or taking my machine learning skills to greater and greater depths in this final capstone.

Although the emotional impact of being accidentally dropped from the course surprised me, it also gave me a new example of false positives and the importance of keeping a human in the loop on automated systems (including ML models). Beyond that, getting re-added to the class lit a fire in me to wrap up the model training work and get everything submitted and wrapped up.

This push to finish a month and a half early was significantly helpful because it caused me to stop gold-plating the final deliverables and allowed me to close a chapter in my educational journey and focus more of my attention on my book and course development projects. This in turn was better for my growth and stress levels.

I loved this class. This is going to be one of my favorite classes from my time at Franklin and it's because I was given the flexibility to go apply my knowledge and learn new things, while being given reinforcing lectures on the concepts already covered in the program.

The only content I'd tweak about this class would be having the body text of the discussion prompts be more than a single vague sentence, because some of them are really not clear in what they're looking for at all just from the initial paragraph prior to the action items.

Great course. Thank you again.