Julian Popovski-Jones, James Chang and Dylan Hoi at University College London have been working with Avanade and Microsoft on a project to unmask “black box” AI algorithms.

The primary goal of their project is to create a web-base application for explaining how a machine learning model makes certain predictions. The application would enable anyone to upload machine learning models and datasets and receive concise explanations, or to upload them to a database where they can be shared with the general public.

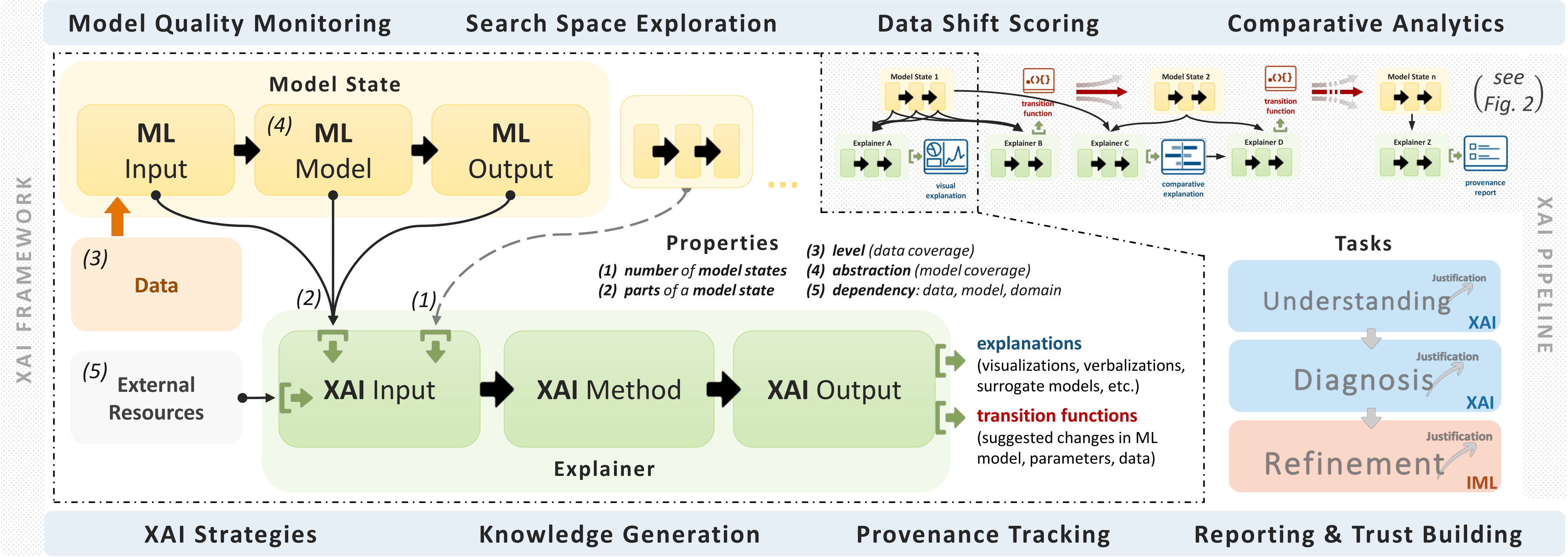
The challenge to realise this goal is enormous. While ideas for an online AI explainer is not rare, there is no actual related product, meaning the team has no references to review.

Subsequently, the team faced another issue: none of the team members have extensive knowledge in the field of AI technology, meaning the team would have to start from researching and learning the fundamentals of machine learning.

Aside from these issues, there are still countless challenges ahead - which explanation module should they choose for their product? How to host the web-based application? How much is it gonna cost to maintain this application? Nevertheless, the team remained confident that they will be able to achieve all of their primary goals.

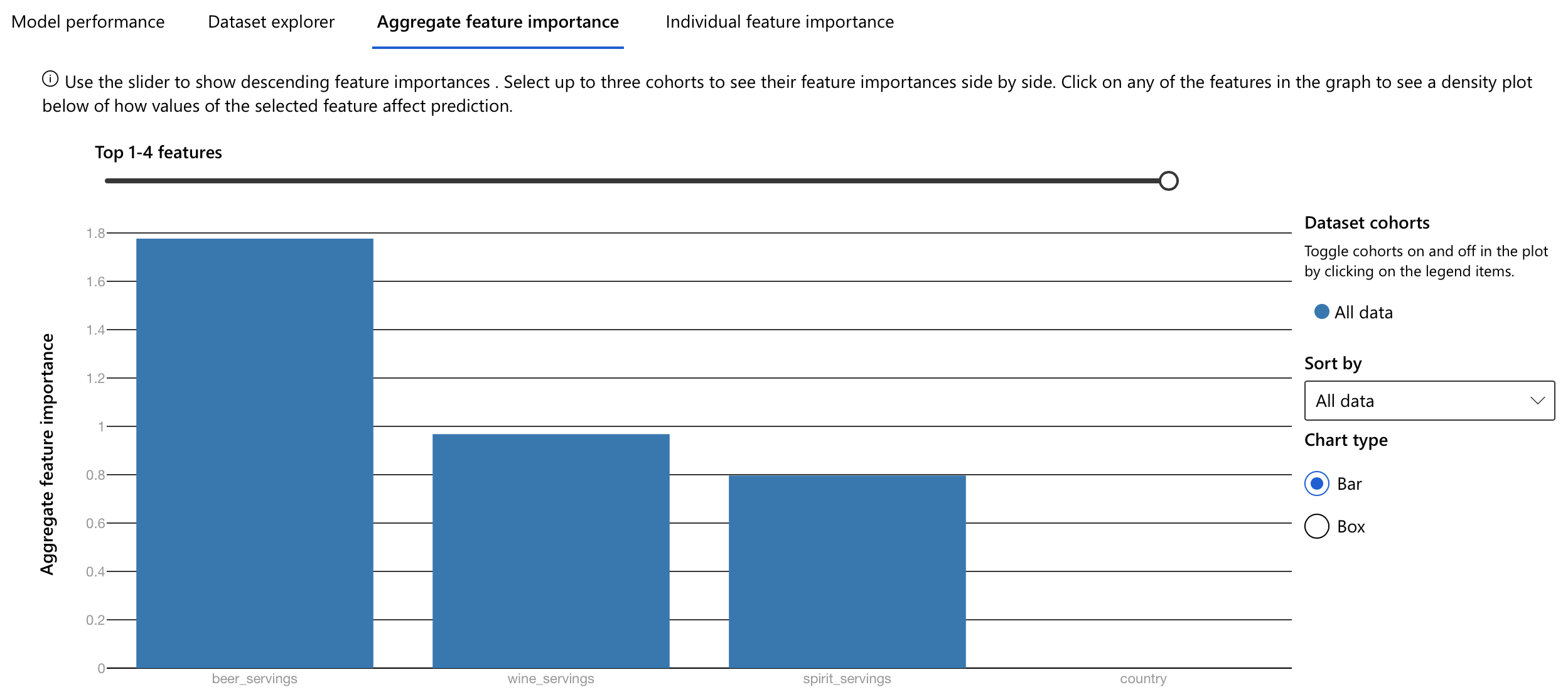
The team started by reviewed two products related to explainable AI in order to come up with a solution to the application: Microsoft Azure’s machine learning explainer, and explAIner.

explAIner is a framework for explainable AI and Interactive machine learning that users can use to incorporate explainer with trained models. Though an AI explainer service itself, explAIner provided a good picture of how machine learning models work and connects with explainer tools.



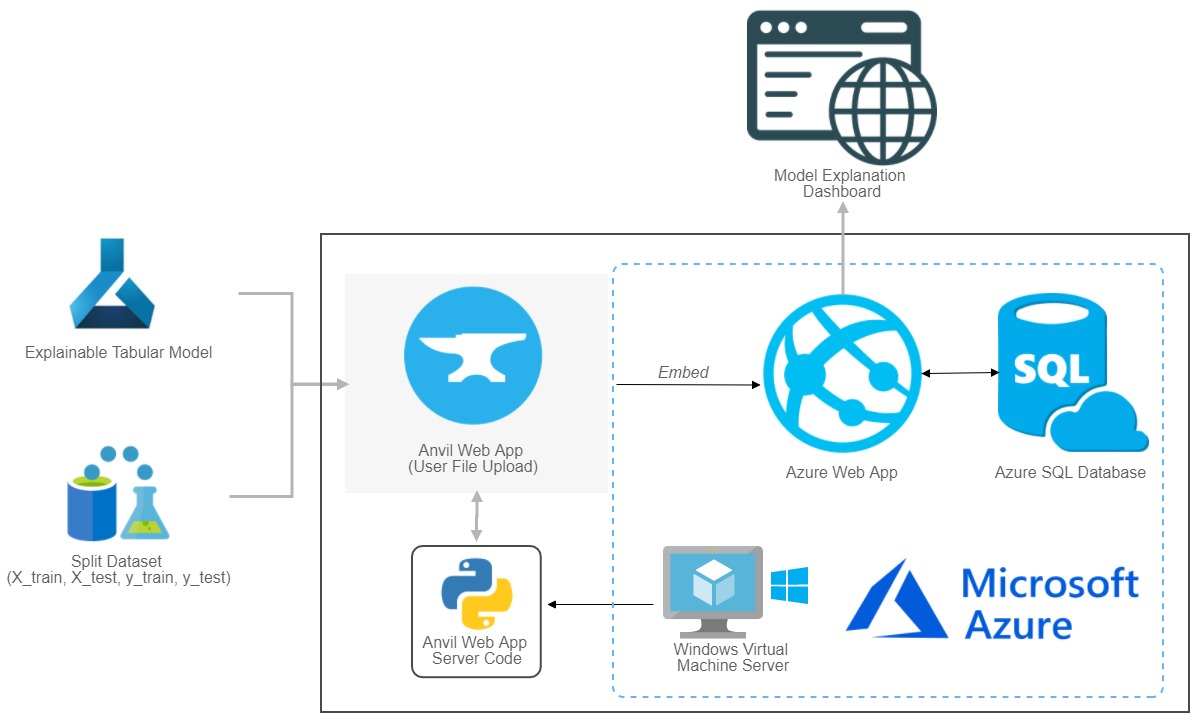
XAI framework diagram from explAIner

Microsoft Azure’s machine learning explainer is a built-in functionality inside Azure Machine Learning Studio, where it can either be used as a standalone module or submodule of their Automated ML tool. The team is able to learn how an explainer should explain models concisely from Azure’s machine learning explainer.

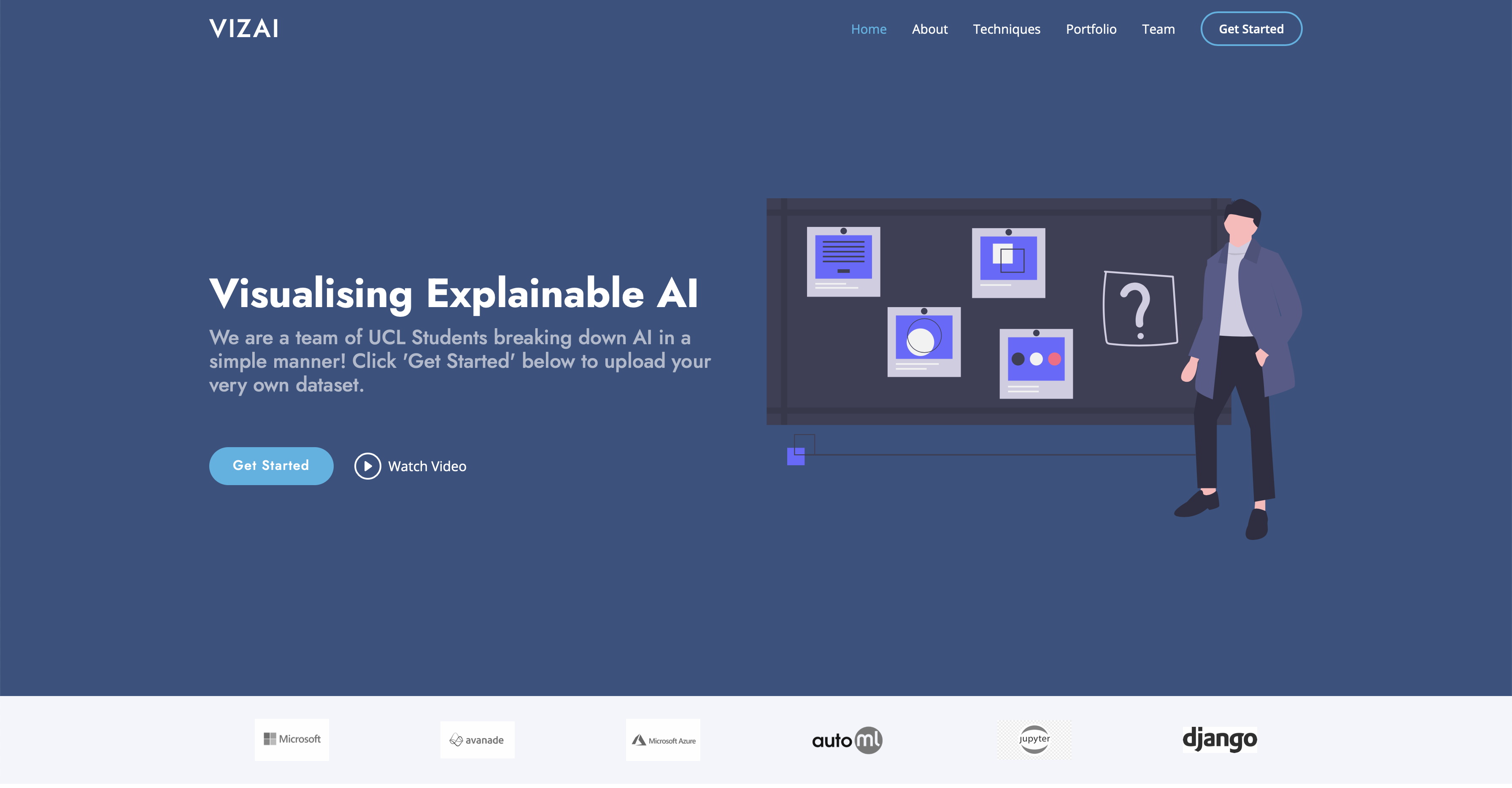


Explainer module explaining a model’s prediction bias

The solution the team came up with is an online explainer application that meets all of the primary goals, costing less than $100 a month to maintain. It is hosted on Microsoft Azure’s virtual machine and database services. In its core is an Azure powered SHAP explainer module, while its frontend websites are built with design and template tools like Anvil, Django and Bootstrap.



VizAI’s system architecture diagram



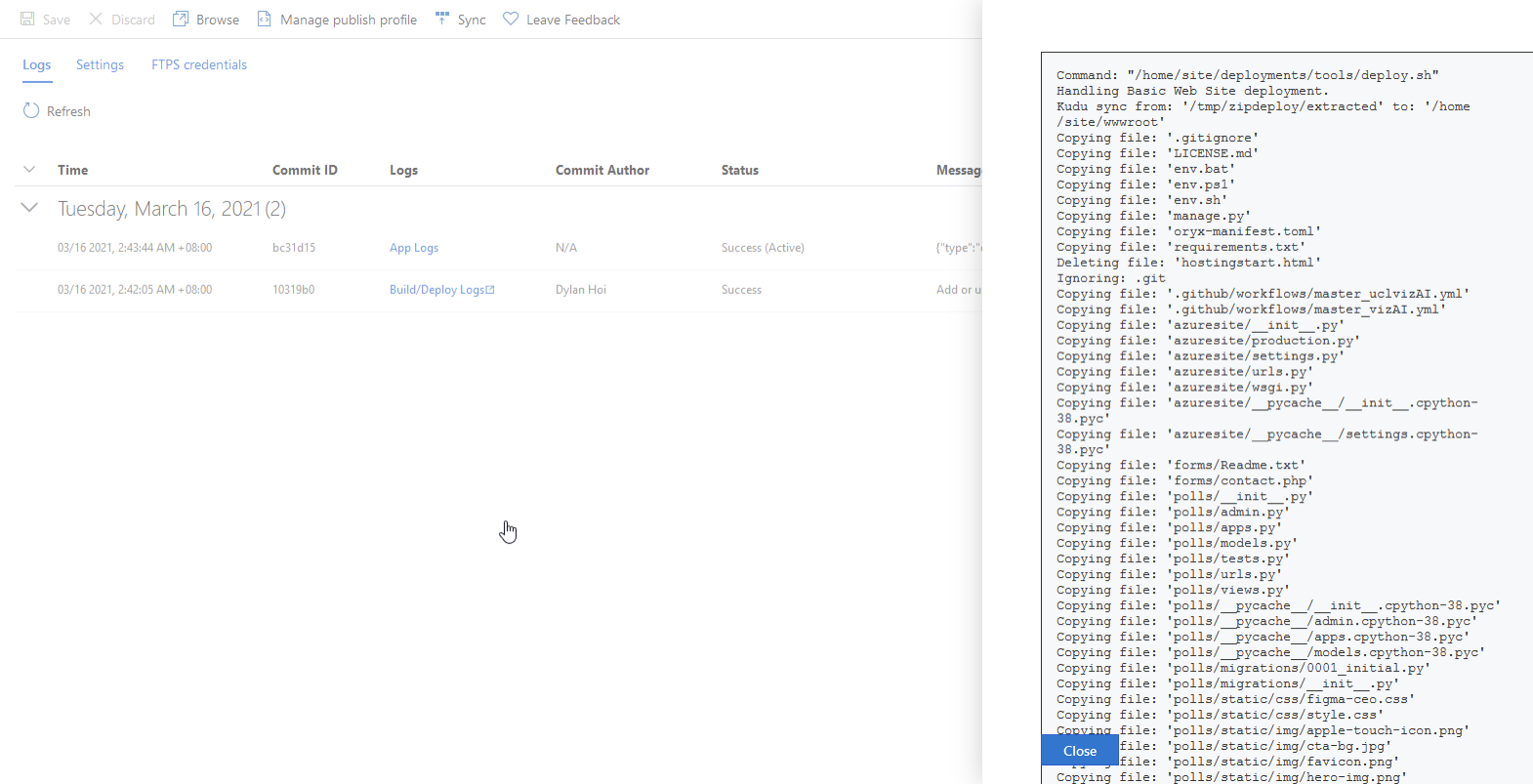
Prototype website

The most important technologies and libraries used in this project are the Microsoft Azure service and the Azure powered SHAP explainer.

Microsoft Azure is responsible for hosting Both the frontend websites and backend explainer module. Its tight integration with Azure’s SQL database service also means data storage can be easily dealt with by connecting the services together.

Azure powered SHAP explainer module is an explainer based on Game Theory. It estimates he Shapely Value for every data point and assigns each feature as a player, then attempts to distribute the summation of predictions across all players to show how much each feature contributed to the overall prediction. This made it particularly strong in explaining the popular tree-based machine learning models.

In the process of building the prototype, the team also utilised Github Action’s CI feature, creating a pipeline for building and testing the application. Once in place, the team no longer needs to manually test and build every new feature, saving precious time for the team.



GitHub Action’s CI feature in action

The team has compiled a list of Learning Points from this project:

* Observe carefully. There may not be similar examples to reference to, but precious knowledge can still be learnt by studying individual parts of various loosely related projects.
* Don’t reinvent the wheel twice. Using reliable pre-built tools and libraries can be both faster and more economic.
* Communication is the key. With good communication through every stage of development, workload can be distributed more evenly and progresses made faster.

Future works on the project includes developing more in-depth functions for more technically advanced users and upgrading system backend to handle more robust machine learning models. The team aims to continue the development of VizAI explainer, and welcomes anyone willing to contribute to make explainable AI easier and faster.

GitHub link: https://github.com/deCourier/vizAI.git