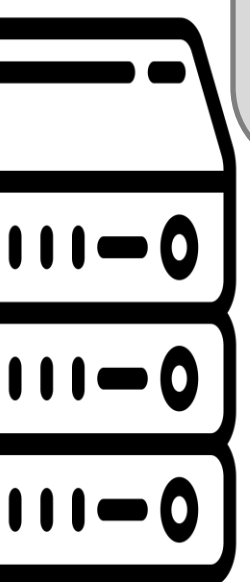


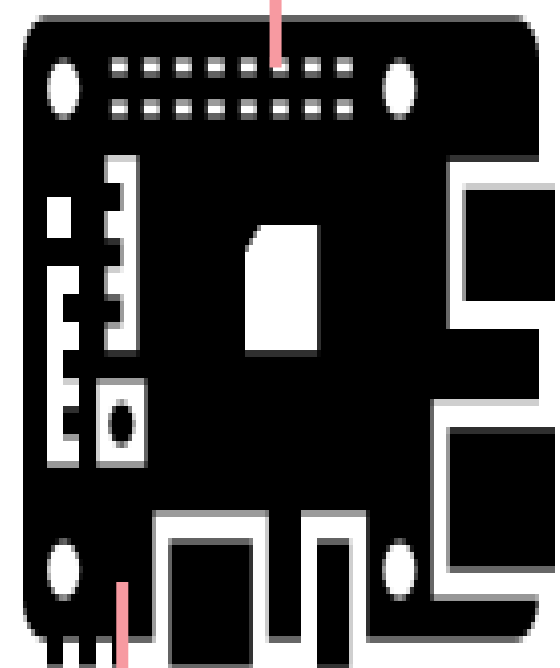
**Introduction/Background/Motivation**

Did you know that the pig harvesting industry can involve a lot of statistical analysis? Gro Master's mission is to help farmers can predict exactly when their pigs will be appropriate weight for market. As it turns out, pigs can actually be too heavy for market and get penalized (being worth less), while pigs that are too light for market are a waste of potential profit. Farmers need to be able to control their pig weights, as well as easily view and predict them. Gro Master helps them accomplish this with their mix of hardware and software. At the farms, the sorters record pig weights and automatically assigns pigs to different feeding pens. Their web app helps farmers view their pig statistics and predict market time as early as possible.

When the pi has no internet, new data is kept in a local Postgres server. We also experimented with a file-watching system in Rust.



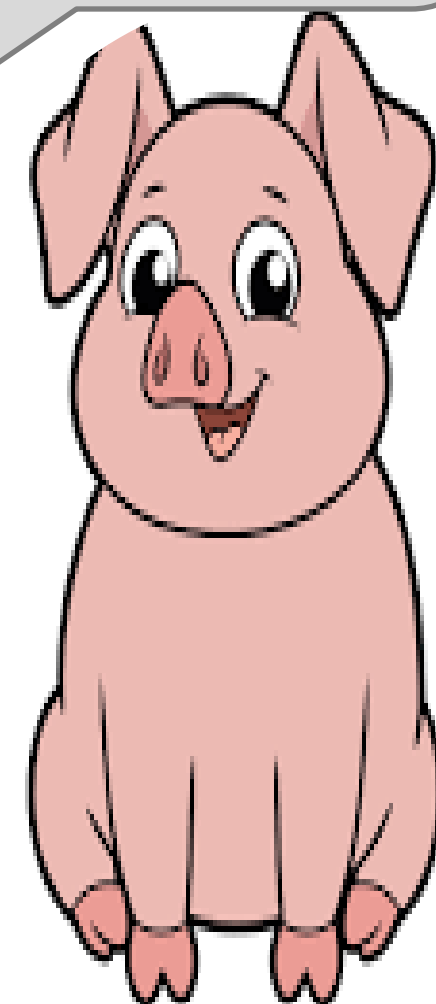
Data is synced between the local Postgres server and the central server ASAP.



Heavy Pig



When pigs enter the sorter, they are weighed, and allowed to enter a specific feeding pen based on their target weight. Smaller pigs, for example, get assigned to more feed.



Small Baby Pig

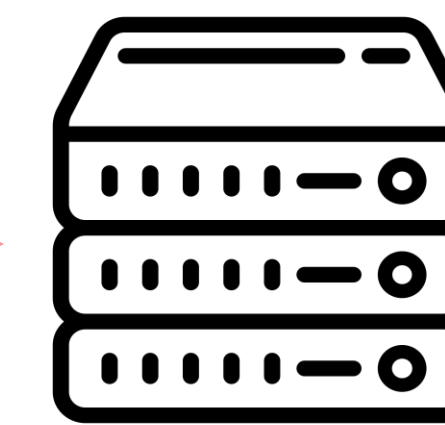
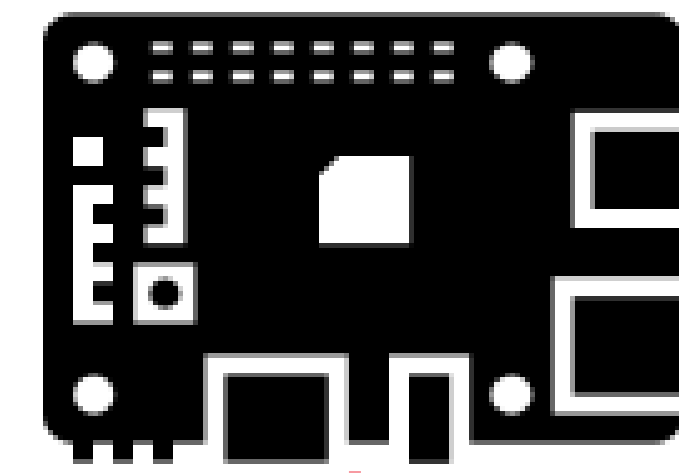
When the pi has internet, data is sent directly to central Postgres server

**HARDWARE**

Farmers also requested an app that let them remotely control their pig sorters. The pig sorters have several basic functions, originally handled on-site with the raspberry pi hardware:

- On/off
- Open/close gate
- Other proprietary techniques

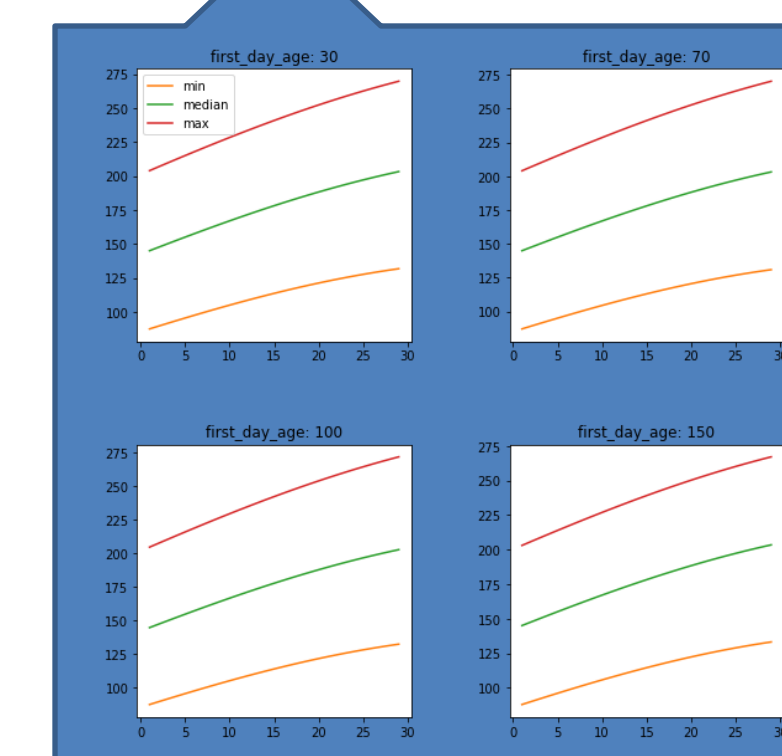
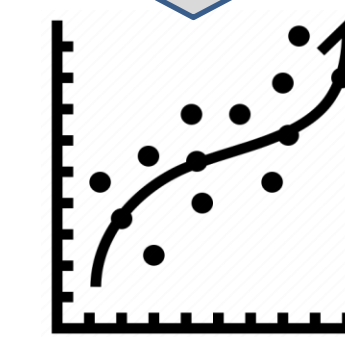
We incorporated this functionality into interactive forms in the new website



Originally, our team fit Multi-Linear models for each psuedopig. The image below is us estimating fits for different pig ages. Our team also experimented with LSTM and ARIMA models for forecasting, though the data was too noisy to make conclusions.



Flask



All data from several farms' pig weights is stored in a PostgreSQL instance on AWS

- Developed python methods to fit models directly on fresh PostgreSQL data
- Resolved an issue in which our regression models were fitting on incorrect data



Created python script to automatically insert weather records for each farm into a database table

Our team merged the codebase of the previous barebones Flask server and a new one we designed with the AppSeed Flask App Generator. specifically

- Reviewing old code to understand how the Flask server pulls current data and authenticates users
- Designing the UI for the new website with specific features and decisions that would appeal to clients
- Debugging our local server staging environments to be able to comfortably develop and test
- Creating the flask server (Creating new routes, Decoupling Flask templates and using new CSS
- Integrating Javascript graphing APIs to display live data for farmers



**Conclusion and Acknowledgements**

Gro Master's top priority for the future should be securing more clients and obtaining more complete real-world data. They simply lack clean, consistent, and labelled data that's necessary to test and improve their forecasting. This would require acquiring more clients and creating a formal TOA to gather their data. They have been doing a great job promoting their sorting machines and developing specialized models for pig weight forecasting. We've developed a more maintainable codebase and implemented customer feedback into our website. We thank our mentor BJ, Kevin, and Dr. Schinckel for guiding us throughout the various tasks and challenges of this project.

**SOFTWARE**