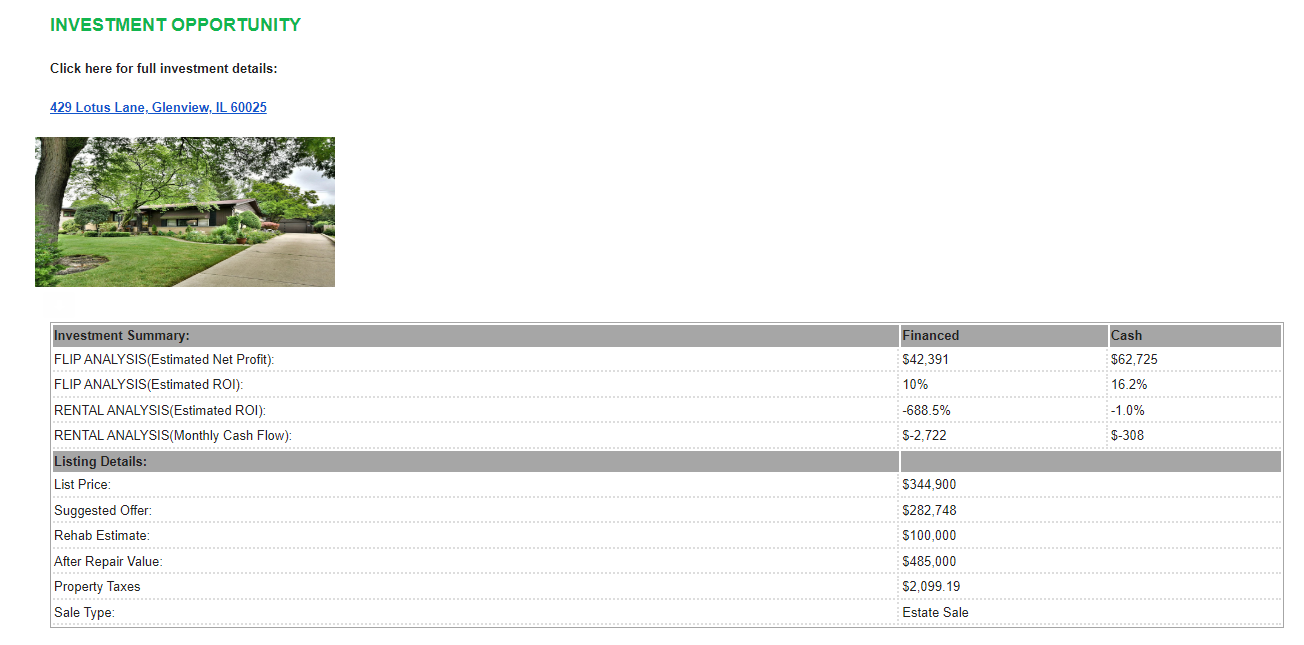
10 Week Demo Video Documentation

Tim Steed

MSDS 434 Fall 2022 Quarter

For my final project I chose to build an application focusing on single family home values. I think the primary difference between my application and some of the others is that mine will (hopefully) present immediate real world use for myself and (again hopefully) will provide financial benefit in the near term. For context, outside of my nine to five job, myself and one of my siblings have an LLC where we purchase, rehab, and sell single family homes in the Chicagoland area. From mid 2020 until very recently our biggest problem was finding viable properties. The housing market was so competitive that homes that projects we were interested in were selling before we had the chance to look at them and do our underwriting to determine realistic construction costs and after rehab values. One of our solutions to this was to subscribe to a service called ‘ClickInvest’ used the MRED (the local MLS branch) API to pull properties from the MLS, identify them as potentially good investments, do some very basic underwriting, and email them out to their users to evaluate (and even submit offers electronically). The idea was it lets you evaluate a lot more deals than you would be able to do in traditional methods like filtering the MLS, Redfin, Zillow, etc. on certain criteria, in addition to having a speed advantage (you could submit an offer on a property within minutes of the listing agent submitting it to the MLS. I was never able to backwards engineer how exactly they were identifying these properties (I suspect it was some type of clustering methodology based on price per square foot, price per bedroom/bath, etc. for a given municipality). Their emails look like this:



Some of the metrics here would be easily replicated based on variables like the current rate I can borrow at and the projected hold time. Others like the rehab estimates I think I could probably improve upon because I have done many similar projects in the past, and can from experience, give a better estimate of how much it will cost to for example, take a bathroom down to the studs and renovate, than their estimates which seem to be largely based off the square footage. In fact, the MLS has a property description required where there are fields like heat source, roof age, etc. and if I can set up a similar system of scraping their data via API I think I could set up logic that includes really granular rehab estimates (for example, I typically replace radiant heat with a furnace and ductwork when I do these projects, or if the furnace is over 15 years old, keep the ductwork but replace the furnace), and could project these costs based on the value in the ‘heat source’ field on the MLS. The after repair value should be pretty easy to replicate with a regression model. I don’t have access to the MLS, so can not access that data yet, but I am in the process of get access and it should only take another few weeks. The purpose of my project was really just to get a version 1.0 of this functionality where I can programmatically read property data and predict its value based on property specific variables. The ones with the largest dollar difference between list price and predicted value grouped by zip, filtered to zips that I am actually interested in (I can only really handle projects that are within ~30 minutes of my home) would be candidates for investment. I could set up a batch process to pull this data daily, push it in csv form to my github repository, then set up a trigger in the GCP pointing to my github repository that runs the analytics whenever there are changes to the repository, and email me the top candidates daily. The cost of ClickInvest’ was $2500/yr, so when I can successfully get this application running it will be a great way to save money and improve the product. There are plenty opportunities for improvement for this application, the version I built for the project is really just the first step in what I am sure will take me a few months to complete. First of all, the real estate data provided in the GCP analytic hub was not free, they only provided a small sample dataset for no cost, so I elected to go with a real estate data set I found on Kaggle. The problem with the Kaggle data is it was exclusively properties on the east coast, which I would not be able to commute to in order to rehab. I will need to get real market data from municipalities near where I live, which will require me to become aan MLS member, which takes a few months. I will have to rework the code to utilize the MLS API and push an updated csv to my github repository daily. The data from Kaggle only had 8 features outside of the target to train my model on, so I will also have to train a new model in Vertex AI to handle real MLS data which I am assuming has hundreds of features. Considering how easy Vertex AI is to use, I don’t really see this as a problem though, more of an opportunity to access a superior predictive model. One technology I would like to incorporate into future iterations of the project is Docker to separate some of the functionalities of the application into microservices, not necessarily because this application requires it, but more because I want to learn more about Docker and this would provide an opportunity to get practical real world experience.