BEEKIN

Private and Confidential: By accessing the link below you are agreeing to keep the information provided private and confidential. Any intellectual property created or provided is the sole property of Beekin, Inc (the 'Company'). No assignment will be accepted without express written permission of the President of the Company. By agreeing to complete this exercise you are waiving your right to Intellectual Property created before or during this process.

This link contains datasets of listing rents from different websites. All listed properties are from Idaho, United States.

Note that you can assume the following:

- - "beekin_id" is a hashed entity. Each unique combination of address, number of beds and bath and sqft are mapped to a unique identifier called "beekin_id".
- - Each "beekin_id" can appear more than once in the dataset. This happens when the same unit is listed more than once on one or more websites, at multiple timestamps. - Each listing/row of data has a "posted_at" attribute. This represents the date when the listing was posted.
- - The dataset has been cleaned beforehand, e.g. there won't be any negative values in price/market rent

Background

Here is the full context to the business use case, and what you (as a senior data scientist at Beekin) have done so far to translate the problem into an ML solution:

- - Sales and product manager wish to create a solution that provides reliable market rent estimates given a user inputted information
- - You have already demonstrated a proof of concept to the internal stakeholders using an Xgboost model, data that was available until the end of 2021 and evaluating model performance with the metric median absolute percentage error. The training variable was transformed by taking the z-normalization of the log of the price.

- - You have used the following columns to train the xgboost model: “latitude”, “longitude”, “sqft”, “beds”, “baths”, “property_type”, “posted_at”. “property_type” is label encoded into four categories (Townhouse:0, Single Family (per building):1, Multifamily (many apartments in one building):2, Condo:3). “posted_at” is a timestamp from when the record was recorded, and is not directly used in the model. However, a derived field, the number of days between the posted_at date and the first of January, 2014, is used.

Base model

You can view this model in the file Proof_of_concept.ipynb (also accessible in the above link).

Goals

The goal is now to determine how much the model can be improved. To help with this, you have two new data sources that were not available at the time the original model was created: - Partial data from 2022, which is of the same format as the original data set.

- - Census data on block group level (an area smaller than zipcode) on some of the population statistics. In particular, ‘population’ records the number of people living in that block group, ‘housing’ is the number of households, ‘education’ is the number of people that have completed high school and ‘transportation’ is the number of commuters in that block group.

Of course, improvements to the model that do not use this data can also be considered.

Deliverables:

You will need to implement an experiment tracking and versioning package to log information such as model parameters, chosen datasets, and other metadata that will enable you to recreate the settings that generated each model, for all the deliverables mentioned below. (Feel free to pick one of the following: MLflow/ Neptune/ Amazon Sagemaker/ Verta AI/ Azure Machine Learning/ Comet/ Weights & Biases)

1. -You should perform some experiments to determine if the census data is adding any signal to the model. Use the experiment tracking and versioning package to log all relevant information.
2. **A 1-page memo on**: feature risks (which features theyd personally exclude even if it improved accuracy), temporal design (how frequently to refresh, how to model for volatile markets), failure tests (how to detect silent degradation), monotonicity trade-offs (accuracy versus monotonicity).
3. Your product manager wants to make sure that the model will refresh periodically, preferably as soon as new data is ingested into the data warehouse. Write a script that will allow you to refresh the model as soon as new data is available in the data warehouse (you can use the 2022 data as an example). Also create a simple diagram for prod architecture sketch : Data ingestion, Validation, Training, Model registry, Deployment, Monitoring, Rollback
4. You would like to monitor the change in performance between each model refresh. Decide on the range of acceptable differences in model performance and write a function to monitor each model update.

Submit the responses (with relevant links if needed) to **hiring@beekin.co**. You are expected to spend no more than 48 hours on this task, as the lift of a data scientist at Beekin is a balance between velocity and veracity.

Please note, this email is not able to respond to queries about the process or the assignment.