Beekin Rent Model — Design & Risk Memo

Purpose

The goal was to test whether adding census-level demographic features could improve the rent-estimation model. I compared two versions: **(1)** a baseline model using only property and location features, and **(2)** an enhanced version that also included census features. Both were evaluated using Median Absolute Percentage Error (MdAPE).

During testing, the baseline model performed better than the census-enhanced model, which means the census variables did not add meaningful signal in this dataset.

Feature Risks

Some features can make a model look more accurate in the short term, but create long-term risk:

- **Demographic and socioeconomic signals** (education, population composition, commuter density) can unintentionally introduce bias or fairness concerns. Even if they improve accuracy, I'd prefer not to rely on them for pricing decisions.
- **Features tied to listing behavior** rather than true market effects (duplicate postings, repost frequency, platform noise) can inflate performance but won't generalize well.
- **Highly unstable or region-specific features** increase maintenance cost and make the system harder to monitor and explain.

In this context, the small accuracy change does not justify the added risk.

Temporal Design

The model should refresh on a regular cadence, but the frequency should depend on market conditions:

- **In normal conditions:** quarterly retraining is sufficient.
- **In volatile periods** (rapid price swings): monthly retraining with closer monitoring.

Validation should follow a forward-in-time approach to avoid using future information. Performance should be reviewed across time slices, not only at the aggregate level, to catch drift earlier.

Failure Tests

To reduce operational risk, I'd monitor:

- Changes in MdAPE between model versions with a tolerance threshold.
- Performance by region, property type, and size brackets, not just globally.
- Predicted-to-listed-rent ratios over time to catch calibration drift.
- Automatic rollback to the previous model if performance drops beyond the tolerance window.

The goal is early detection before issues reach users.

Monotonicity Trade-offs

Some relationships should remain directionally consistent — for example, larger units should not be predicted cheaper than smaller ones, all else equal.

Adding soft monotonic constraints may slightly reduce raw accuracy, but it improves interpretability, user trust, and product safety, which is often the better trade-off.

Conclusion

In our experiments, adding census features slightly worsened performance compared to the baseline model. Given the small signal gain, the added bias risk, and the operational complexity, **I would not recommend deploying the census-augmented version**.

The baseline model is more stable, easier to explain, and a stronger foundation for future improvements.