Forecasting Neighborhood-Level Rent Change in the Chicago Metropolitan Area Using Multi-Source Socioeconomic and Amenity Signals

Max Gillum*

Department of Computational Mathematics, Science and Engineering

Michigan State University, East Lansing, MI 48824

(Dated: November 11, 2025)

(THub: https://github.com/maxgillum4/cmse492_project)

Abstract

This project develops a machine learning framework to forecast short-horizon changes in residential rent at the ZIP code level in the Chicago metropolitan area. Rather than estimating static price levels, the objective is to predict the next-period rent index and year-over-year rent growth by leveraging a hybrid formulation that combines time-series dynamics with cross-sectional predictors. The primary data source is Zillow's Observed Rent Index (ZORI), which provides monthly rent estimates by ZIP code. ZORI is fused with socioeconomic covariates from the U.S. Census Bureau, labor market indicators from the Bureau of Labor Statistics, and amenity density derived from the Yelp Fusion API via spatial assignment of businesses to ZIP polygons. The methodological plan compares three model families of increasing complexity, beginning with regularized linear regression to establish a transparent baseline, followed by random forests to capture nonlinearities, and gradient-boosted trees for state-of-the-art tabular performance. Evaluation emphasizes temporally appropriate validation that prevents leakage by training on historical slices and testing on out-of-sample months. The expected contributions include a reproducible data fusion pipeline, an empirical assessment of the relative importance of socioeconomic versus amenity features, and geographically interpretable forecasts of rent growth. Preliminary exploration confirms adequate coverage across urban and suburban ZIP codes and reveals meaningful correlations between prior rent changes, income, and amenity density.

BACKGROUND AND MOTIVATION

Rental housing affordability is a salient urban policy concern. Stakeholders ranging from planners and nonprofit organizations to property managers and prospective residents benefit from early indicators of neighborhood-level rent acceleration or deceleration. Traditional hedonic models typically estimate contemporaneous price levels and may not generalize when the underlying drivers shift, particularly across neighborhoods with heterogeneous amenities and labor markets. This study advances a predictive approach centered on forecasting change rather than level. By combining historical rent trajectories with cross-sectional socioeconomic conditions, local labor market indicators, and amenity density, the model targets next-period rent growth as an outcome that is aligned with real decisions such as budgeting, siting, and risk monitoring. The Chicago metropolitan area offers sufficient vari-

ation in neighborhood characteristics to make model comparison informative while keeping data acquisition tractable. The contribution of this work is a transparent, reproducible pipeline that measures out-of-sample forecast accuracy, quantifies the marginal importance of interpretable drivers, and produces spatially coherent maps that communicate risk to nontechnical stakeholders.

DATA DESCRIPTION

The primary target variable is Zillow's Observed Rent Index (ZORI) aggregated at the ZIP code level with monthly frequency. ZORI is constructed by Zillow Research from rental listings and quality controls and is widely used for rent trend analysis. Socioeconomic predictors are drawn from the U.S. Census Bureau's ZIP Code Tabulation Area statistics, including median household income, population, and age structure. Local labor market indicators, notably unemployment rates, are incorporated from the Bureau of Labor Statistics and mapped to ZIP codes via standard ZIP–county crosswalks. Amenity density is computed from the Yelp Fusion API by retrieving business locations within the metropolitan bounding box, assigning each business to its containing ZIP polygon through a spatial join, and aggregating counts by category. The working panel comprises [**N_zips**] ZIP codes across [**T**] months after filtering to ZIPs with at least [**M**] consecutive observations. Initial exploration shows [**brief note on missingness/outliers, e.g., "short gaps in ZORI filled with 3-month rolling means"**]. Two figures summarize key properties: a distribution of year-over-year rent changes indicating [**mean/variance/skewness**] and a correlation matrix showing associations among prior rent changes, income, amenity density, and the target. Planned preprocessing includes standardizing continuous predictors, one-hot encoding categorical features if introduced, imputing short gaps via rolling means, and constructing lagged features such as prior monthly and annual rent changes.

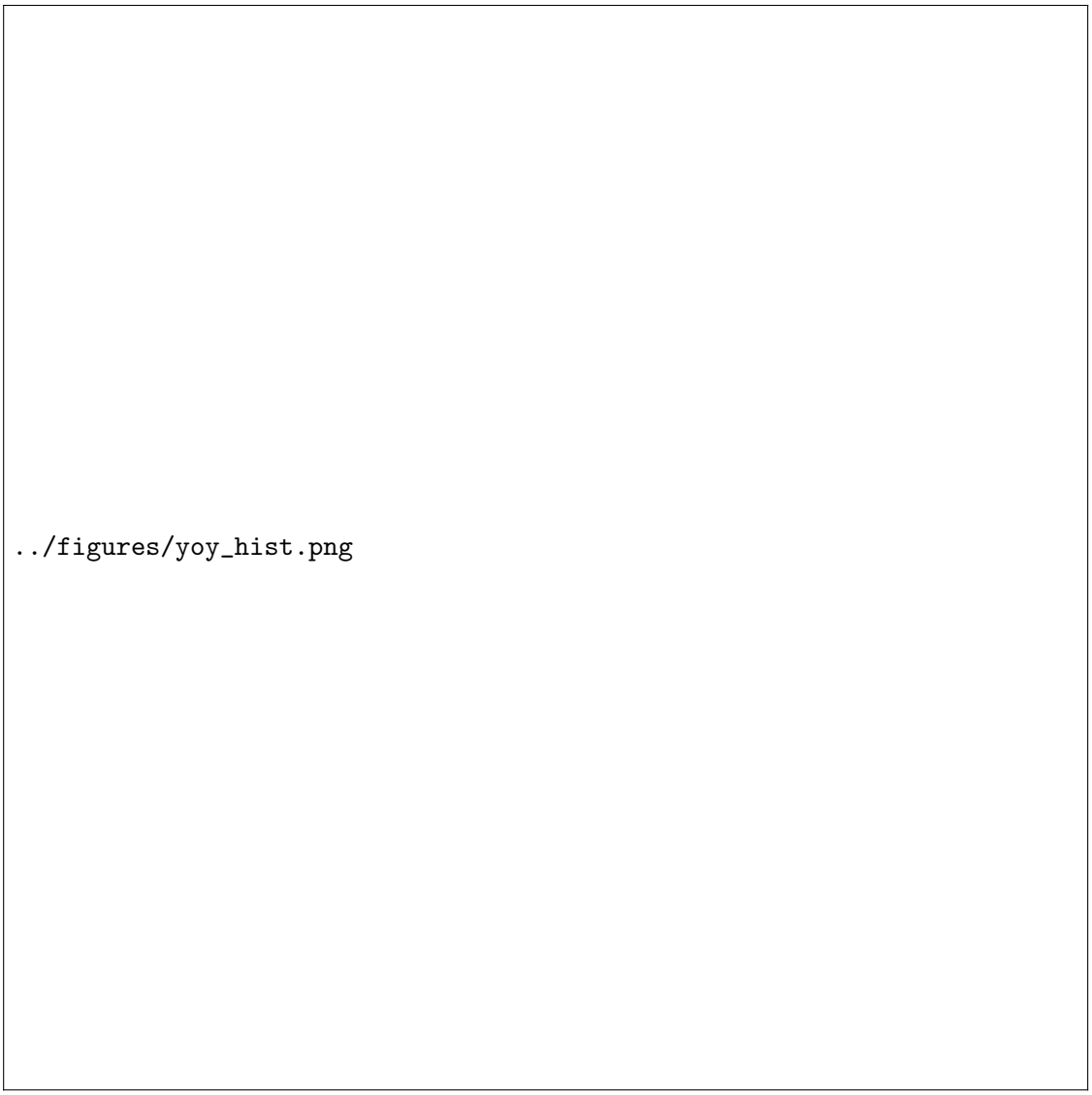

FIG. 1: Distribution of year-over-year ZORI changes across Chicago ZIP codes.

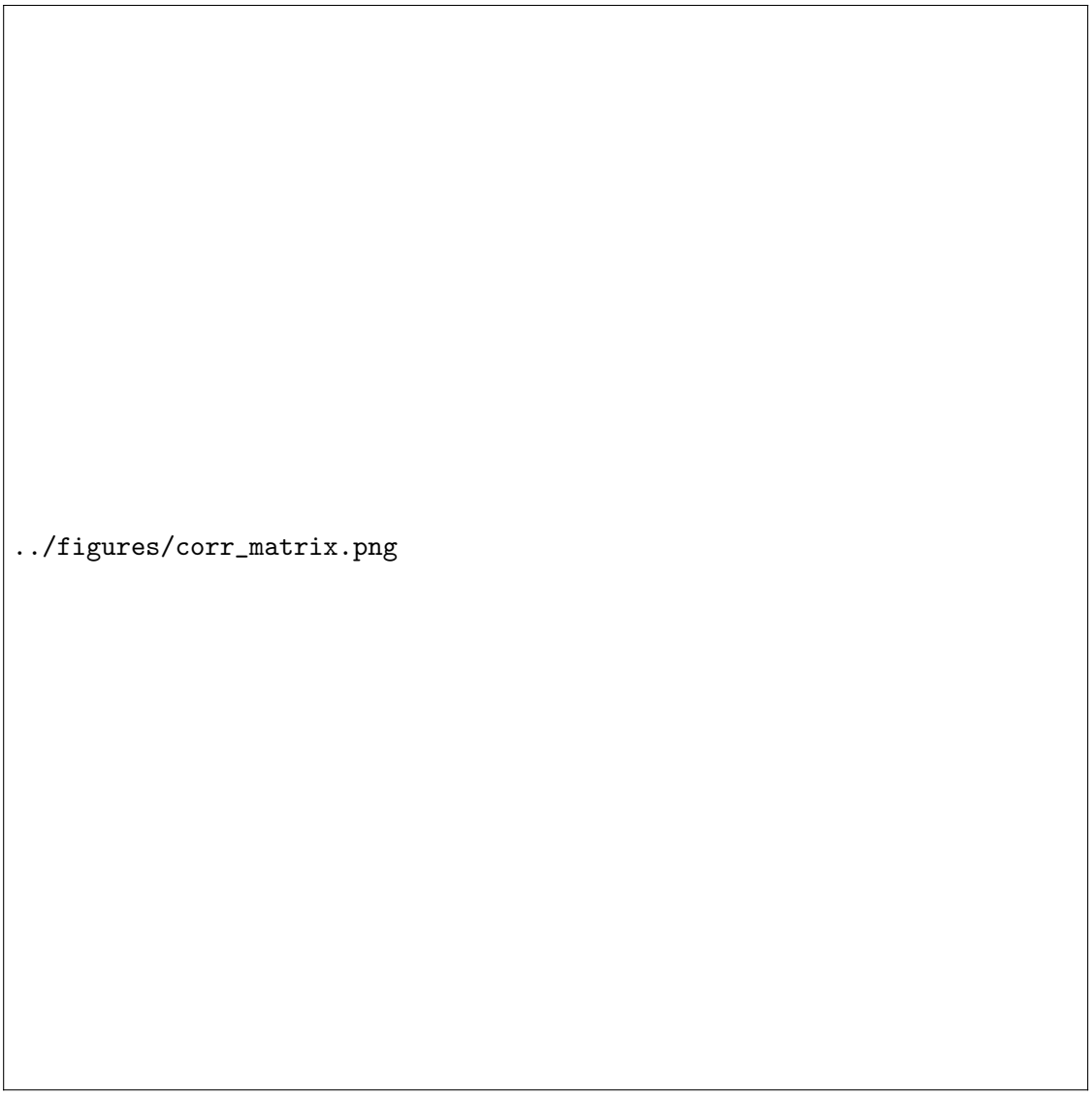

FIG. 2: Correlation matrix among key predictors and target variables.

PREPROCESSING

Preprocessing proceeds in four stages. First, monthly ZORI is reshaped from wide to long format and merged with ACS ZCTA features and county-level BLS unemployment rates by constructing a ZIP–county mapping from the HUD crosswalk. Second, continuous features are standardized within the training window, while categorical indicators are one-hot encoded if required. Third, short missing spans are imputed using rolling means that only reference prior months to avoid leakage. Finally, lagged features are created, including one- and twelve-month rent changes, as well as rolling-window summaries that stabilize noisy

signals. All transformations are implemented in scikit-learn pipelines to ensure identical treatment across training, validation, and test splits.

Data Splitting

Time-aware splitting prevents leakage. Training includes all months up to year [**T_0**], validation comprises [**T_0+1**], and testing uses [**T_0+2**] (e.g., train through 2023, validate on 2024, test on 2025 YTD), with a grouped split by month to ensure each fold is strictly forward-looking.

Feature Engineering

Feature engineering adds lagged ZORI levels and differences, rolling statistics to capture momentum and mean reversion, population-normalized amenity densities, and interactions such as income-by-amenity that may signal gentrification pressure. All features are aligned to ensure the prediction at month $t + 1$ only uses information available at or before month t.

Scaling, Transformation, and Encoding

Continuous features are standardized using parameters fit on the training set only. Right-skewed count variables (e.g., amenities) may be log-transformed after adding a small constant. Any categorical fields introduced by crosswalks are one-hot encoded. Imputation and scaling are encapsulated in a pipeline to avoid data leakage.

MACHINE LEARNING TASK AND OBJECTIVE

The task is supervised regression that forecasts next-period rent change at the ZIP level. Humans and simple rule-based approaches struggle to integrate heterogeneous signals across neighborhoods and to update beliefs dynamically as conditions shift. Machine learning provides a principled way to fuse historical trajectories with socioeconomic and amenity covariates while controlling for overfitting through cross-validation and regularization. The

target is next-month rent index (level) and year-over-year change, with models trained to minimize squared error and evaluated on temporally held-out months.

MODELS

Three model families are compared in increasing capacity. A regularized linear model (Ridge) offers a high-bias, interpretable baseline that handles correlated predictors and yields stable coefficients. A random forest regressor introduces nonlinearities and interactions with robust performance on tabular data and provides permutation-based importance. A gradient-boosted tree model (XGBoost) serves as the primary high-accuracy benchmark with tuned depth, learning rate, and subsampling that typically excels on structured features.

Model Selection

Models are evaluated under the same temporal split and feature pipeline. Hyperparameters are tuned by grid or randomized search on validation months with early stopping where applicable. Final models are refit on train+validation and reported on the untouched test months.

Model 1: Ridge Regression

Ridge provides a transparent baseline and mitigates multicollinearity among socioeconomic variables. It defines a linear decision surface and is resilient to moderate noise.

Model 2: Random Forest Regressor

Random forests capture nonlinear feature interactions and are robust to outliers. They provide intuitive global importance via permutation and support partial dependence analyses.

Model 3: XGBoost Regressor

Gradient boosting builds an ensemble of shallow trees to reduce bias and variance. With learning-rate and depth control, it typically achieves strong accuracy on structured, mixed-scale inputs.

Regularization and Hyperparameter Tuning

Ridge uses an L_2 penalty with λ selected on validation months. Random forest depth, number of trees, and minimum leaf sizes are tuned to balance bias and variance. XGBoost tunes maximum depth, learning rate, number of estimators, subsampling, and column subsampling, with early stopping on validation months to prevent overfitting.

TRAINING METHODOLOGY

Training minimizes squared error subject to the temporal split. Let $y_{i,t}$ denote the rent-change target for ZIP i at time t, and $\hat{y}_{i,t}$ the model prediction. Ridge minimizes mean squared error with an L_2 penalty on coefficients; tree ensembles minimize squared error through their inherent objectives. Learning curves are monitored by plotting training and validation errors over time-sliced folds to ensure stability and to select early-stopping rounds for boosting.

Loss Functions

For Ridge,

$$\mathcal{L}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{w}^\top \mathbf{x}_i)^2 + \lambda\|\mathbf{w}\|_2^2. \tag{1}$$

Random forests and XGBoost optimize squared-error objectives at each split; XGBoost further regularizes tree complexity via depth and leaf penalties.

Training Process

Each model is trained on the training window, tuned on the validation window with temporal cross-validation, and evaluated on the test window. Preprocessing pipelines are

fit only on training data and applied to later periods to avoid leakage. Learning curves and residual diagnostics are saved as figures for inclusion in results.

Model Summary Table

Table I summarizes model classes, key hyperparameters, and regularization.

TABLE I: Summary of models, parameters, and training methodology.

Model	Parameters	Hyperparameters	Loss Regularization
Ridge	$\mathbf{w}, b$	λ	MSE L_2
Random Forest	Trees	depth, n_estimators, min_samples_leaf	MSE Implicit via averaging
XGBoost	Trees	depth, lr, n_estimators, subsample, colsample_bytree	MSE Tree + early stopping

METRICS

Root mean squared error (RMSE) is the primary metric to quantify average forecast error magnitude on the test months, while mean absolute error (MAE) provides a robust companion less sensitive to large residuals. The coefficient of determination (R^2) is reported for explanatory power. Spatial choropleths and predicted-versus-actual scatterplots complement numeric metrics by revealing geographic coherence and rank-order fidelity across ZIPs.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \qquad \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{2}$$

RESULTS AND MODEL COMPARISON

Results will compare baseline, Ridge, random forest, and XGBoost on the held-out test months. Performance tables will include RMSE, MAE, and R^2, alongside plots of predicted versus actual outcomes and spatial maps of residuals. Discussion will address accuracy, stability across neighborhoods, and computational costs, and it will justify the selected best model for the deployment-like scenario of monthly nowcasting.

Performance Comparison

TABLE II: Model performance metrics on test months.

Model	**RMSE**	**MAE**	R^2
Ridge	[ ]	[ ]	[ ]
Random Forest	[ ]	[ ]	[ ]
XGBoost	[ ]	[ ]	[ ]

Computational Efficiency

TABLE III: Training and inference time for each model.

Model	**Train Time**	**Inference Time**	**Hardware**
Ridge	[ ]	[ ]	CPU
Random Forest	[ ]	[ ]	CPU
XGBoost	[ ]	[ ]	CPU/GPU

Analysis and Discussion

Analysis will explain why particular models perform best, how lagged signals and amenity densities contribute to accuracy, and where errors concentrate. Failure modes such as abrupt shocks will be identified and potential remedies proposed.

MODEL INTERPRETATION

Model interpretation uses permutation importance to quantify global driver strength and partial dependence or accumulated local effects to visualize marginal relationships. These tools clarify how income, unemployment, and amenity densities influence predicted rent changes and help validate that signals behave reasonably across the observed range.

Feature Importance

Tree-based permutation importance will be reported and contrasted with linear coefficients from Ridge to cross-validate directional effects.

Model Behavior Analysis

Partial dependence plots for top features will be inspected for monotonicity, thresholds, or interactions that align with economic intuition.

CONCLUSION

This work proposes a practical forecasting pipeline for neighborhood rent change that integrates time-series momentum with socioeconomic, labor, and amenity signals. The best-performing model will be selected based on temporal generalization and interpretability. Limitations include reliance on ZIP-level aggregation and potential sparsity in amenity coverage; future work will explore tract-level modeling, richer mobility features, and transfer to other metros.

Summary of Findings

The study aims to deliver accurate, interpretable monthly rent-change forecasts and a clear ranking of driver importance that can inform planning and budgeting.

Limitations and Future Work

Key limitations involve data gaps, boundary effects at ZIP edges, and sensitivity to sudden shocks. Future work includes finer spatial units, additional labor and mobility signals, and probabilistic forecasting.

Final Remarks

The resulting codebase and figures are designed to be reproducible and extensible to additional regions such as Columbus, OH and East Lansing, MI.

I thank the CMSE 492 teaching team for guidance. This project is part of CMSE 492 at Michigan State University.

---

* gillumma@msu.edu

[1] Zillow Research, "Zillow Observed Rent Index (ZORI)," https://www.zillow.com/research/ (accessed 2025).

[2] U.S. Census Bureau, "American Community Survey (ACS)," https://www.census.gov/programs-surveys/acs (accessed 2025).

[3] Bureau of Labor Statistics, "Local Area Unemployment Statistics (LAUS)," https://www.bls.gov/lau/ (accessed 2025).

[4] Yelp Developers, "Yelp Fusion API," https://docs.developer.yelp.com/ (accessed 2025).

Additional Figures and Tables

Additional supporting material will include residual maps, PDP/ALE plots, and ablation study tables.

Code Availability

The complete code for this project is available at: https://github.com/maxgillum4/cmse492_project