Department of Computer Science

Value Visionaries

Linn Kloefta
Areeb Eshan
Bhavya Patel
Nihar Naik

May 4, 2025

Abstract

Predicting how home values change each year is important for investors, planners, and anyone involved in housing decisions. This project focuses on forecasting year-over-year (YoY) home price growth at the ZIP-code level in Georgia. We used Zillow Home Value Index (ZHVI) data along with socioeconomic and demographic information from the American Community Survey (ACS) to build a dataset that captures both housing trends and local conditions.

Each row in our dataset represents a ZIP code in a specific year, with features like volatility, average monthly growth, and previous YoY changes from the ZHVI time series. We also added county-level ACS variables like education levels, poverty rates, unemployment, and household structure to capture broader context.

To find the most useful features, we used two different selection methods: mutual information and f-regression. Then we trained three models — Decision Tree, K-Nearest Neighbors (KNN), and Random Forest — tuning them using GridSearchCV. The models were evaluated using R^2, RMSE, MAE, and SMAPE on a ZIP-based test set. Random Forest performed the best overall, while the other models showed signs of overfitting unless properly tuned.

Overall, we found that recent YoY trends and economic indicators were the most reliable signals for short-term price growth. This setup could be extended in the future to make longer-term predictions as more recent ACS data becomes available.

Keywords: housing prediction, machine learning, ZHVI, ZIP code, ACS

Report's total word count: 3637 words

Contents

List of Figures

List of Tables

List of Abbreviations

ACS	American Community Survey
CAGR	Compound Annual Growth Rate
KNN	K-Nearest Neighbors
MAE	Mean Absolute Error
MI	Mutual Information
RF	Random Forest
RMSE	Root Mean Squared Error
SMAPE	Symmetric Mean Absolute Percentage Error
ZIP	Zone Improvement Plan (ZIP code)
ZHVI	Zillow Home Value Index
YoY	Year-over-Year

Chapter 1

Introduction

1.1 Background

Understanding housing price trends is important for economic planning, smart investing, and making housing more accessible. In the U.S., housing markets can vary a lot — not just between states or cities, but even from one ZIP code to another. Predicting growth at a local level helps everyone from city planners to individual buyers make better decisions.

Zillow's Home Value Index (ZHVI) gives monthly estimates of home values across the country and is a great starting point for analysis. But price data alone doesn't tell the full story. Things like income, education levels, household makeup, and transportation access all influence how housing prices change over time. In this project, we combined ZHVI time series data with demographic information from the American Community Survey (ACS) to build a model that can forecast local home price growth across Georgia ZIP codes.

1.2 Problem statement

The main goal of this project was to predict next-year home value growth at the ZIP-code level in Georgia. A lot of existing models work at a broader level — like metro areas or states — and often leave out important social and economic context. We wanted to build something more detailed, that looks at local trends and uses both housing and demographic data.

This isn't an easy problem. Home prices are sensitive to the economy, and there's a lot of noise at the local level. Some ZIP codes don't have much data, and there are also issues with reporting timelines and inconsistent trends from year to year. Still, our hope was that with a strong enough dataset and good feature engineering, we could make reliable short-term predictions.

1.3 Aims and objectives

Aim: Build a model that predicts one-year-ahead home value growth for ZIP codes in Georgia, using both housing trends and demographic data.

Objectives:

- Load and clean Zillow ZHVI data for Georgia ZIP codes.
- Create features from the ZHVI time series (e.g., volatility, growth rates, prior YoY).
- Add in ACS demographic data by county and year.

- Compare different models: Decision Tree, KNN, and Random Forest.
- Use feature selection (mutual information and f-regression) to reduce dimensionality.
- Tune and evaluate each model using a ZIP-based test split.

1.4 Solution approach

We built a long-format dataset where each row represents a single ZIP code in a given year. The target was defined as the YoY growth from that year to the next — letting the model learn how past trends relate to future outcomes.

From the ZHVI data, we engineered features like:

- Final ZHVI value for the year,
- Average monthly price growth,
- Volatility (standard deviation of monthly percent change),
- CAGR (Compound Annual Growth Rate),
- Number of years with negative YoY growth,
- YoY growth in the previous year.

We merged in county-level ACS data by year, including indicators like:

- Education levels,
- Poverty rate,
- Household structure,
- Vehicle ownership,
- Unemployment,
- Age breakdowns.

After preprocessing, we trained three models: Decision Tree, KNN, and Random Forest. We used ZIP-based train/test splits to prevent target leakage and tuned hyperparameters using GridSearchCV. Feature selection helped reduce noise and focus on the most informative variables.

1.5 Summary of contributions and achievements

What we were able to do in this project:

- Built a cleaned, long-format dataset combining housing and ACS data at the ZIP-year level.
- Created interpretable housing features from raw price data.
- Evaluated and tuned three models with proper train/test validation.
- Identified top predictors like recent YoY growth, volatility, and education levels.

- Made a visual tool to show predictions over time for individual ZIPs.

Our tuned Random Forest model gave the best results on the test set, while KNN and Decision Tree performed well but showed signs of overfitting without tuning. All models were able to pick up on general trends, and model performance mainly came down to how well the feature selection matched the model's complexity.

1.6 Organization of the report

This report is broken up into six chapters. Chapter 2 explains how the data was collected, cleaned, and turned into features. Chapter 3 goes over the modeling process and includes results, tuning outcomes, and visualizations. Chapter 4 discusses what the results mean and what worked well (or didn't). Chapter **??** wraps things up and suggests future directions.

Chapter 2

Methodology

2.1 Overview

This chapter explains how we built the dataset, engineered features from the ZHVI time series and ACS data, and trained and evaluated our models. All work was done in Python using `pandas`, `scikit-learn`, and `matplotlib`. Our focus was on building a predictive setup that avoided target leakage and relied only on information available at the cutoff year.

2.2 Data sources and filtering

We used two main data sources:

- **Zillow Home Value Index (ZHVI)**: Monthly home value estimates by ZIP code, accessed from Zillow's data portal Zillow Research (2025).
- **American Community Survey (ACS)**: County-level demographic and economic data from the U.S. Census Bureau, accessed through their data portal U.S. Census Bureau (2025).

We filtered the ZHVI data to only include ZIPs in Georgia, removed rows with missing city or metro names, and excluded ZIPs with more than 12 consecutive months of missing values. For ZIPs missing metro data, we filled it based on the most common metro for the city or county, unless the city was ambiguous.

The ACS data was pulled for all counties in Georgia. Since it is only available at the county level, each ZIP was assigned the values from its corresponding county for the matching year.

2.3 Rolling dataset creation

Each row in the final dataset represents a ZIP code and a year, using only data available up to the end of that year. We used a rolling window approach: for each ZIP-year pair, we created features from the historical ZHVI series leading up to December 31 of that year. If a ZIP didn't have at least 24 months of usable data, we skipped it.

We interpolated missing values linearly and skipped any year where we couldn't compute the YoY from the prior year. The final YoY value for the *next* year was used as the target, meaning we are predicting one year ahead for each row.

2.4 Feature engineering

2.4.1 ZHVI-based features

From each ZIP's historical ZHVI values, we extracted:

- `FinalZHVI`: home value at the end of the year,
- `AvgMonthlyGrowth`: mean monthly growth in percent,
- `Volatility`: standard deviation of monthly growth,
- `CAGR`: compound annual growth rate since the start of the series,
- `NegativeGrowthYears`: count of past years with negative YoY growth,
- `YoY_LastYear`: YoY growth from the previous year.

The target, `YoY_target`, was the YoY growth from the current year to the next. For example, for a 2016 row, the target would be 2017's growth.

2.4.2 ACS-derived features

Each ZIP inherited its county's ACS values. We engineered the following from raw ACS columns:

- Percent renters, people below poverty, and households with no vehicle,
- Percent with a bachelor's degree or higher,
- Unemployment rate,
- Household sizes (one-person and 4+),
- Age group breakdowns: percent ages 0–17, 18–34, and 65+.

For 2024, we forward-filled values using 2023's ACS data since 2024 estimates were unavailable.

2.5 Final dataset preparation

Before modeling, we grouped rare county and metro names into an `Other` category to reduce sparsity, then one-hot encoded the remaining categorical columns.

To avoid leakage, we split the data by ZIP code rather than by row. That means if a ZIP appears in the training set, none of its years appear in the test set. This ensures the model isn't learning from repeated geographic patterns across time.

We used data from the years 2011 through 2024, excluding 2020 due to unreliable ACS data during the COVID-19 pandemic. Although 2024 lacks observed target values, these rows were included in the final dataset for future forecasting. Only ZIP–year rows with valid targets were used for training and evaluation.

The final dataset had 3,009 rows and 139 columns, including both engineered and encoded features.

2.6 Feature selection

We used two methods to rank features:

- `f_regression`: measures linear correlation between each feature and the target,
- `mutual_info_regression`: detects both linear and non-linear dependencies.

We used the rankings to train models using the top k features for k from 1 to 50, comparing performance using 3-fold cross-validation.

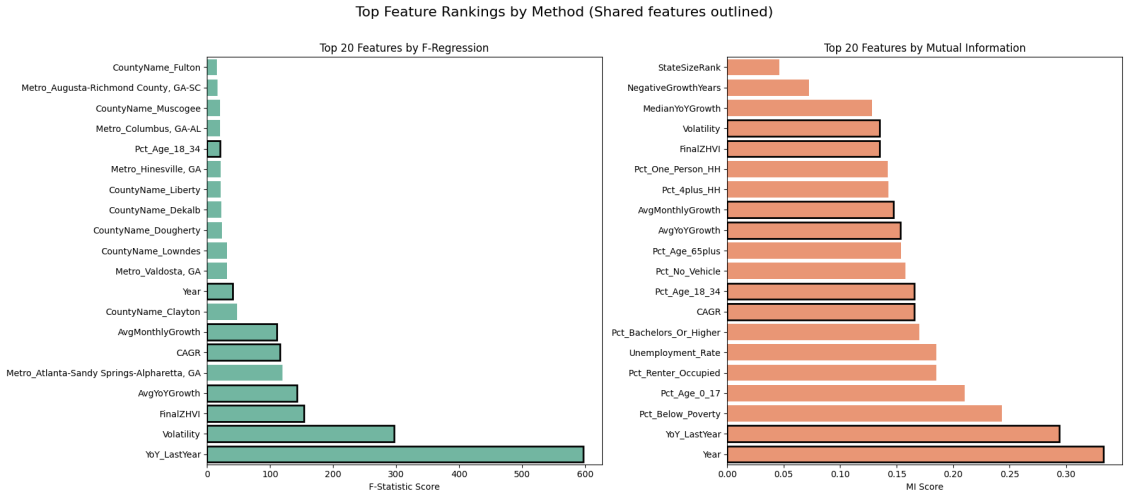


Figure 2.1: Top 20 features by F-Regression and Mutual Information.

Figure 2.1 shows that features like `YoY_LastYear`, `Volatility`, `FinalZHVI`, and `AvgMonthlyGrowth` were among the most informative. Mutual information also highlighted ACS features like poverty and education. These results suggest that both price history and local demographics contribute to future home value trends.

2.7 Model training and evaluation

We tested the following regression models:

- Decision Tree,
- K-Nearest Neighbors,
- Random Forest.

Each model was tuned using `GridSearchCV`. We evaluated performance using:

- R^2,
- RMSE,
- MAE,
- SMAPE.

Performance was reported on the held-out test ZIPs using both metrics and visual inspection of predictions.

2.8 Ethical considerations

The project uses only public datasets with no personal or identifiable information. All data is aggregated at the ZIP or county level. The models are for research and exploration only, and are not meant to support real-world financial decisions.

2.9 Summary

We filtered, cleaned, and reshaped housing data into a rolling ZIP–year format, added engineered time series and ACS features, and carefully split the data by geography to avoid leakage. After ranking features with two methods, we trained and evaluated three models using a consistent pipeline to understand what factors drive next-year home value growth.

Chapter 3

Results

This chapter presents the results of our model training and evaluation. We tested Decision Tree, K-Nearest Neighbors (KNN), and Random Forest regressors to predict next-year home value growth using a ZIP-based test split. The following sections show how model performance varies with feature selection, test accuracy, and behavior over time.

3.1 Feature selection and model comparison

We used mutual information and f-regression to rank features, then evaluated model performance using the top-k features for $k \in [1, 50]$. Figures below show the 3-fold cross-validated R^2 scores across feature counts.

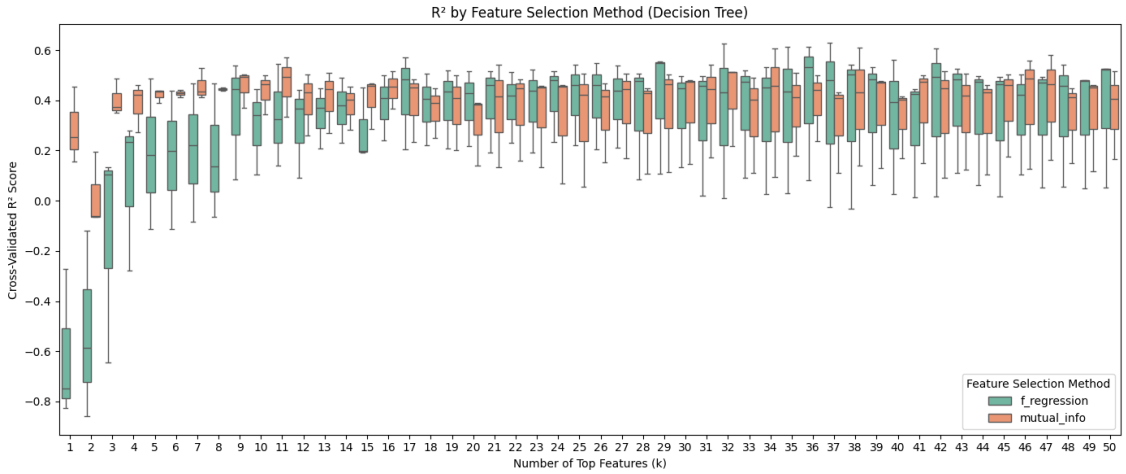


Figure 3.1: Feature selection performance for Decision Tree

Decision Tree improves up to about 8–10 features, then rapidly declines. This is especially evident when using mutual information, which adds non-linear but noisy variables. The model is sensitive to irrelevant input and prone to overfitting.

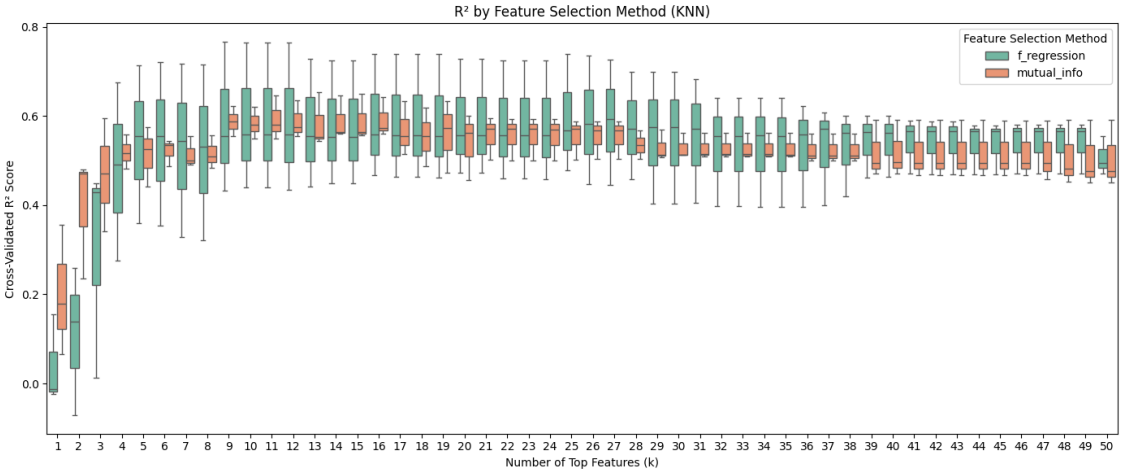


Figure 3.2: Feature selection performance for K-Nearest Neighbors

KNN performance increases up to 9 features, then plateaus. f-regression works best at low k, while mutual information stabilizes later. This shows that KNN benefits from smaller, more targeted feature sets and becomes unstable with excess input.

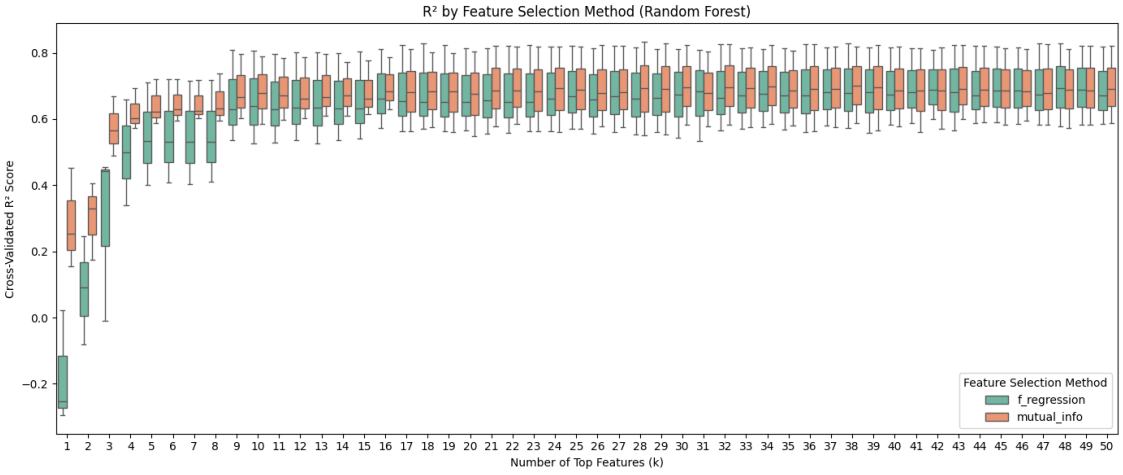


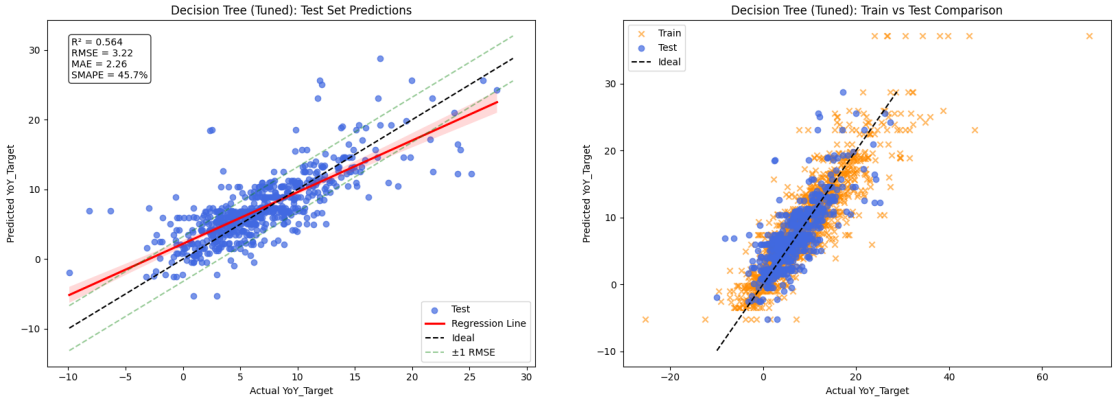
Figure 3.3: Feature selection performance for Random Forest

Random Forest improves steadily up to about 16 features and holds steady beyond that. Its low variance and tolerance for redundant input reflect the model's robustness and internal feature selection behavior.

3.2 Model prediction performance

This section evaluates how well each model predicts on unseen ZIP codes. For each model, we compare test set accuracy and generalization behavior by plotting predicted vs actual YoY values and train vs test predictions.

3.2.1 Decision Tree

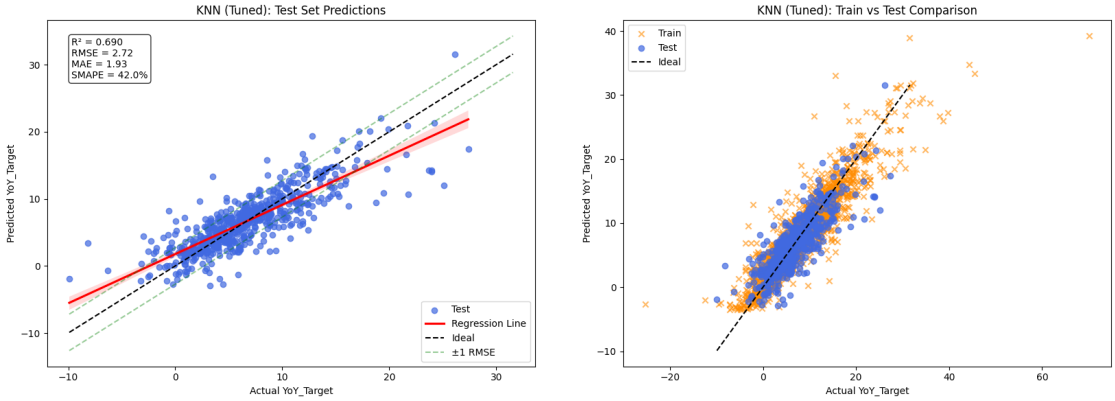


(a) (b)

Figure 3.4: Decision Tree test performance

Figure 3.4b shows that the Decision Tree model compresses predictions toward the center of the target range. High actual values are underpredicted, and low actual values are overestimated. The regression line is noticeably flatter than the ideal diagonal, indicating a strong mean-reverting tendency. In Figure 3.4a, the test predictions fail to reflect the wider spread in actual values, while training points follow the trend more closely. This reflects overfitting on the training set and poor generalization to unseen ZIPs, which is consistent with the model's low R^2 and high error rates.

3.2.2 K-Nearest Neighbors (KNN)



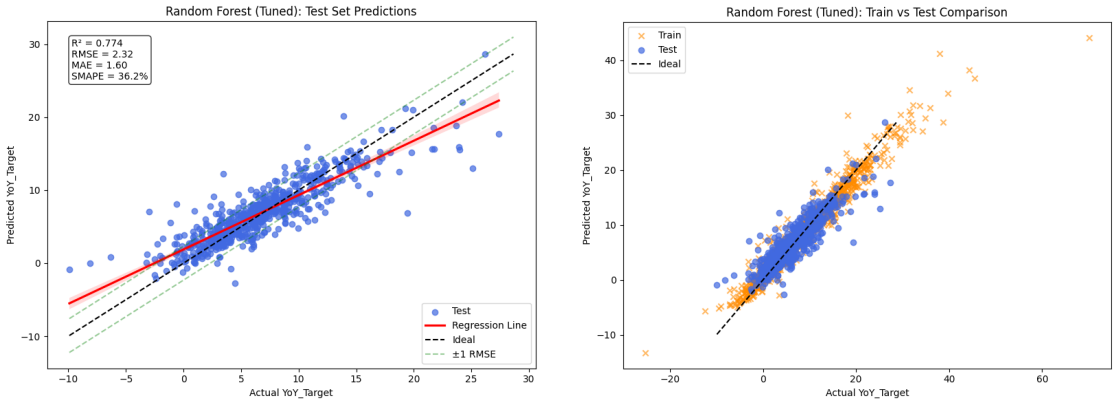
(a) (b)

Figure 3.5: KNN test performance

KNN shows a more balanced prediction pattern. In Figure 3.5a, most predictions fall near the regression line, though there is underprediction at the upper end and more scatter at the

extremes. The central trend is captured more accurately than in Decision Tree. Figure 3.5b shows that both train and test points follow the general diagonal trend, but test predictions are more dispersed. The model performs best when ZIPs with similar history exist in the training data, but becomes less reliable in areas with unique or high-growth patterns.

3.2.3 Random Forest



(a) (b)

Figure 3.6: Random Forest test performance

Figure 3.6a shows that Random Forest predictions generally follow the upward trend, with the regression line reasonably close to the ideal $y = x$ line. While predictions are less dispersed than those of the other models, there is still noticeable spread, particularly at the extremes. The RMSE band captures a good portion of the points, reflecting improved but not perfect accuracy. Figure 3.6b shows that the model performs comparably on training and test sets, with no major signs of overfitting. Overall, Random Forest offers the best balance between accuracy and generalization among the three models.

Table 3.1: Test set performance summary

Model	R^2	RMSE	MAE	SMAPE
Random Forest (16 features)	0.774	2.32	1.60	36.2%
KNN (9 features)	0.690	2.72	1.93	42.0%
Decision Tree (8 features)	0.564	3.22	2.26	45.7%

These results confirm that Random Forest was the most accurate and stable model across unseen ZIP codes, with the lowest error rates and highest R^2. KNN performed competitively but showed more variance on the test set. Decision Tree lagged behind due to overfitting and an inability to model the full target range.

3.3 Prediction trends over time

To evaluate how models perform across years for specific ZIP codes, we selected three ZIPs with high, median, and low ZHVI in 2024. The following figures compare actual vs predicted

YoY growth over time.

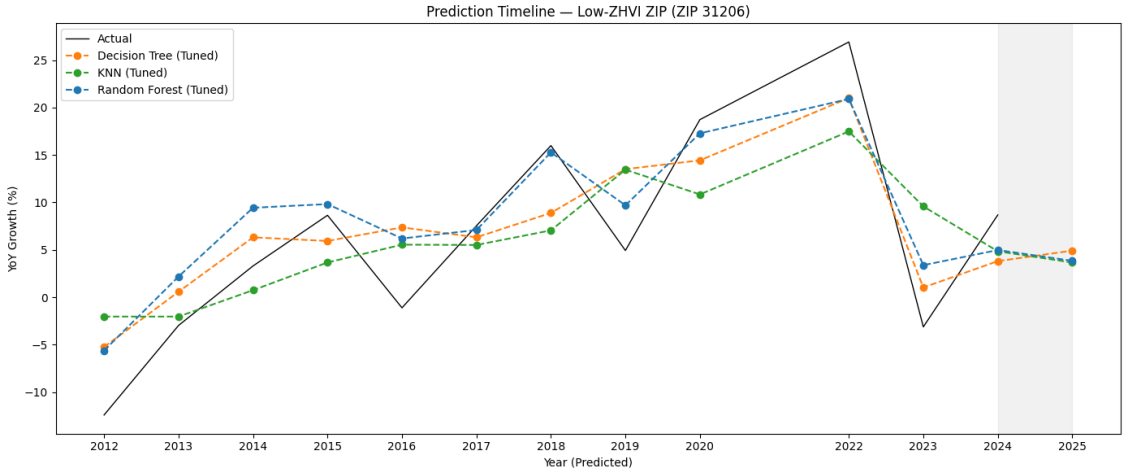


Figure 3.7: Prediction trends over time for low-ZHVI ZIP

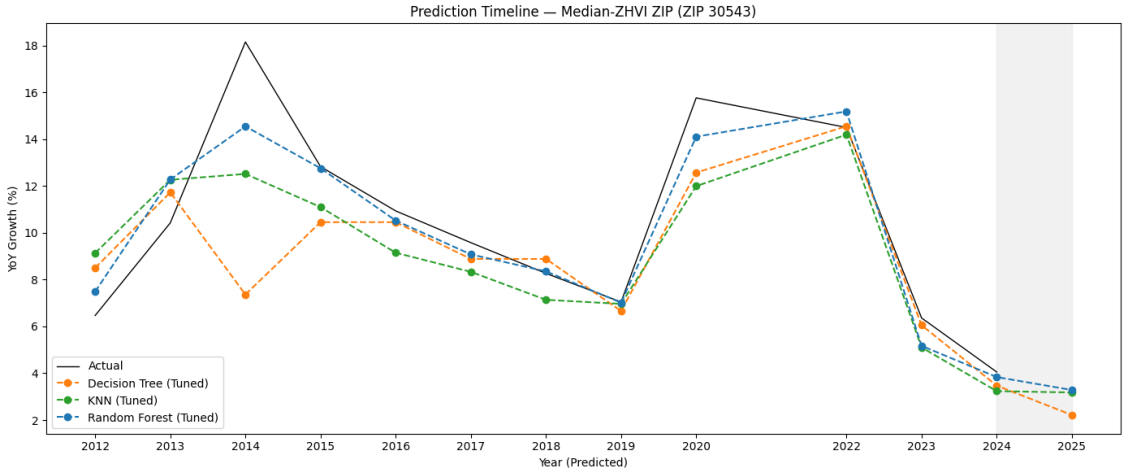


Figure 3.8: Prediction trends over time for median-ZHVI ZIP

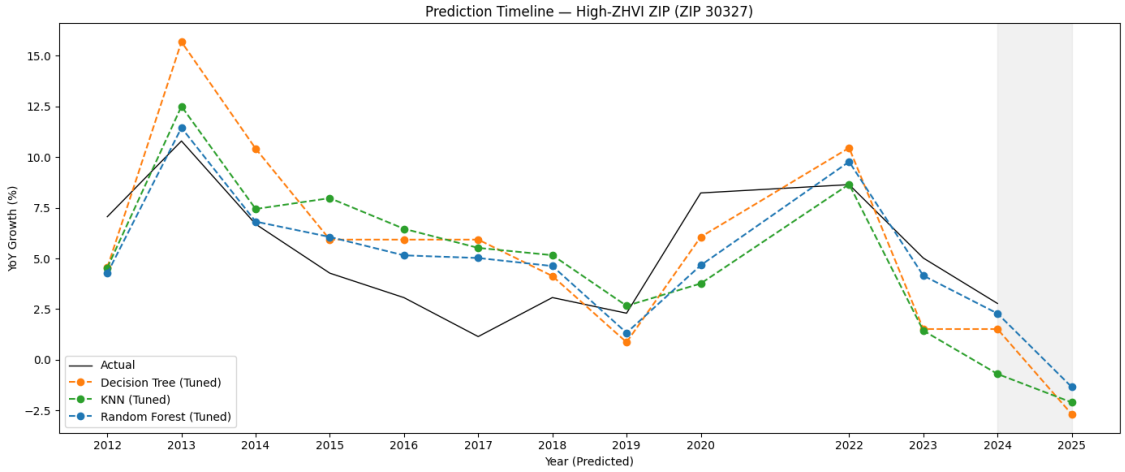


Figure 3.9: Prediction trends over time for high-ZHVI ZIP

Across all three ZIP codes, the Random Forest model most consistently follows the actual YoY growth trend, particularly during years with noticeable dips or inflection points (e.g., 2016, 2019, 2023). In contrast, Decision Tree often smooths over or delays these changes, while KNN tends to flatten predictions and underestimates high-variance years. All models converge in the forecast period (2024–2025), but Random Forest shows closer alignment with the actual data in years prior, especially during periods of volatility.

3.4 Summary

All three models learned meaningful patterns from the data, but Random Forest consistently performed the best. It generalizes well across ZIPs, tracks yearly growth over time, and balances fit without overfitting. KNN was competitive, especially in areas with stable trends, but struggled with rapid market shifts. Decision Tree was the weakest, showing overfitting in training and poor generalization in both time and accuracy-based evaluations.

Chapter 4

Discussion

This chapter discusses why we saw the results we did in the previous section. It focuses on how feature importance was determined, why the models behaved the way they did, and what this tells us about the dataset overall.

4.1 Why certain features ranked high

From both mutual information and f-regression, it was clear that price trend features were the most predictive. Things like previous YoY growth, average monthly growth, volatility, and end-of-year ZHVI consistently ranked at the top. That makes sense, since recent price momentum tends to carry forward, especially in stable markets. These variables give the model a direct view of how each ZIP has been performing recently.

The ACS variables added useful context, especially for areas where price data alone might not be enough. For example, poverty rate, percent with a bachelor's degree or higher, and vehicle ownership were often ranked highly by mutual information. These features probably helped the model adjust for economic conditions that might affect future growth. In particular, education and income levels likely reflect long-term demand and neighborhood stability, even if they're not directly tied to price changes.

f-regression focused more on linear relationships, so it tended to prioritize the ZHVI-based features more heavily. Mutual information captured a broader mix, including nonlinear patterns between demographic indicators and future growth. This explains why mutual information sometimes ranked ACS variables higher, even though they weren't as strong on their own.

4.2 Why the models behaved differently

The Decision Tree performed the worst overall, and the main reason is overfitting. It tried to split on small patterns in the training data that didn't generalize well. Once we added more than 8–10 features, its performance dropped quickly. This shows that the tree wasn't able to handle noise or less relevant input. Since it makes hard splits, it doesn't handle smooth or complex relationships well, especially when features are correlated.

KNN did better, but only up to a point. It works by comparing test ZIPs to similar training ones, so if a ZIP had neighbors in the feature space with similar patterns, it predicted pretty well. But it struggled when a test ZIP didn't have a close match. It also flattened predictions around the average, which is why it underpredicted at the high end. Adding too many features hurt performance, likely because of the curse of dimensionality. KNN does best with fewer, well-targeted features.

Random Forest was clearly the most stable. It improved as we added more features and didn't overfit as quickly. That's expected since it uses many trees with random splits and samples, which helps cancel out noise. It also handles both linear and nonlinear patterns well. Even when some features were weak or redundant, the model still performed consistently. This explains why Random Forest gave us the best test results and tracked the actual growth trends more closely in the time-based plots.

4.3 What the prediction patterns tell us

Looking at the time-based predictions, the models responded differently to trend shifts. Random Forest was the most responsive to real changes — it picked up on dips and rebounds more accurately. That was especially true in volatile years like 2016 or 2023. KNN smoothed over a lot of those changes and generally stayed closer to the mean. Decision Tree sometimes delayed changes or made jumps that didn't match actual trends.

All models performed better in ZIPs with more stable history. Volatile or sparse ZIPs were harder to predict, and that showed up in the wider spread of test errors. Still, even in those cases, Random Forest was usually closer than the others. It was able to combine short-term trends with economic context to make more reliable predictions.

4.4 Limitations to keep in mind

One limitation is that the ACS data is only available at the county level, so ZIPs within the same county get the same demographic values. This flattens differences between ZIPs that might actually be important. It probably reduced how much impact the ACS variables had in the model. Also, some ZIPs didn't have enough ZHVI history to calculate good features, so they were excluded entirely.

Another thing to consider is that the target is only one year ahead. While this keeps the setup simple and avoids long-term noise, it also means we're only capturing short-term effects. A multi-year forecast might show different patterns, especially for slow-changing markets.

4.5 Summary

Overall, the results make sense given the data. The most important predictors were recent price trends and economic indicators that reflect local demand and stability. Random Forest worked best because it could handle complex patterns without overfitting. KNN did well in some cases but was less reliable overall. Decision Tree overfit unless carefully limited. The feature selection helped simplify the models and made it easier to understand what signals were strongest.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This project set out to predict one-year-ahead home value growth at the ZIP code level in Georgia using housing price trends from Zillow and demographic context from ACS data. The goal was to build a model that could pick up on both short-term market signals and longer-term economic patterns.

We built a structured pipeline that cleaned and combined both datasets, engineered useful features, and tested three models. Random Forest gave the best results overall, handling complex and noisy data without overfitting. It consistently outperformed Decision Tree and KNN, especially on unseen ZIP codes and in years with volatile growth patterns.

The features that mattered most were mostly tied to recent housing trends — like prior YoY growth, volatility, and monthly change — but demographic indicators like poverty and education levels also added value. Our feature selection process helped highlight which signals were reliable, and our ZIP-based test split ensured that models weren't just memorizing locations.

In the end, we were able to show that even with some limitations, combining housing and economic data can give solid short-term predictions at a local level. This kind of setup could be helpful for anyone trying to understand where growth might happen next — whether that's a buyer, planner, or researcher.

5.2 Future Work

There are a few clear next steps that could build on this work:

1. **Add material price data.** Including average construction or renovation costs might help explain changes in home value beyond demand. It could give better insight into supply-side effects.

2. **Look at longer-term growth.** Instead of just year-over-year changes, forecasting growth over 3 to 5 years would help with planning and might reduce noise in the predictions.

3. **Expand to more regions and time periods.** Right now the model is limited to Georgia. Applying it to other states or using older data could test how generalizable it is and reveal regional differences.

4. **Build an interactive dashboard.** A simple tool that lets users explore ZIP-level predictions, filter by region or income level, and compare trends could make this more useful in practice.

This project focused on modeling what was possible using the data available. With better geographic granularity and richer features, future versions could become more accurate, more flexible, and easier to use.

References

U.S. Census Bureau (2025), 'American community survey (acs) 1-year estimates', https://data.census.gov/. Accessed 2025-04-15.

Zillow Research (2025), 'Zillow home value index (zhvi) data', https://www.zillow.com/research/data/. Accessed 2025-04-11.