Department of Computer Science

Value Visionaries

Linn Kloefta
Areeb Eshan
Bhavya Patel
Nihar Naik

May 1, 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: x 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 **??** 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 started with Zillow's ZHVI dataset, which includes monthly home value estimates per ZIP code. We filtered the 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.

We also pulled 1-Year ACS data from the U.S. Census Bureau's API for all counties in Georgia. Since ACS is county-level, we assigned each ZIP the demographic values from its 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 removed ZIP-years with missing target values. 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.

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.

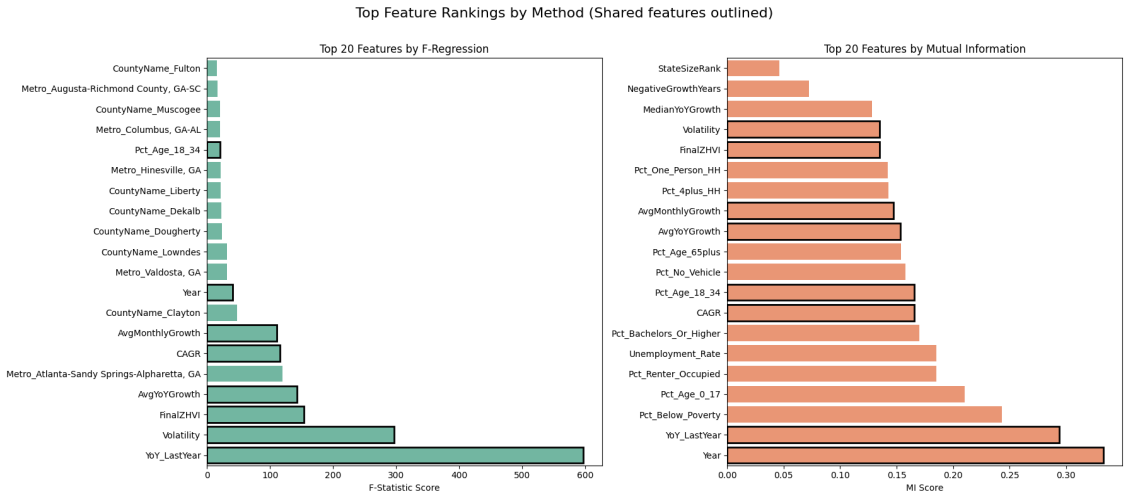


Figure 2.1: Top 20 features by F-Regression and Mutual Information.

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.

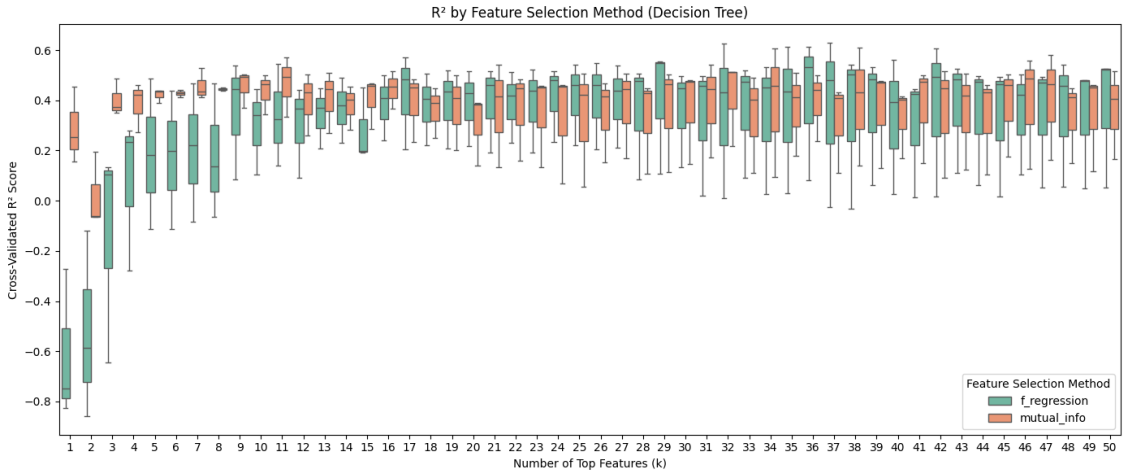


Figure 3.1: Feature selection performance for Decision Tree

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.

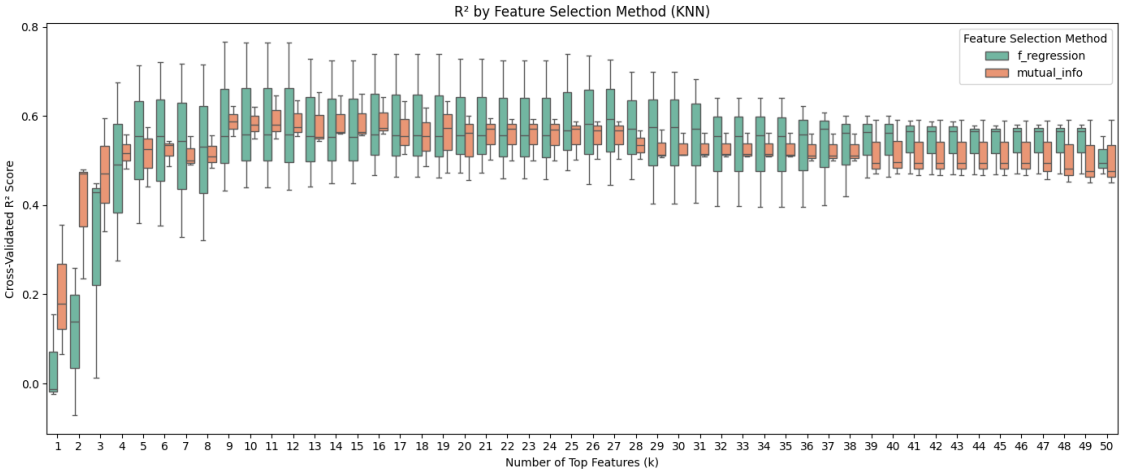


Figure 3.2: Feature selection performance for K-Nearest Neighbors

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.

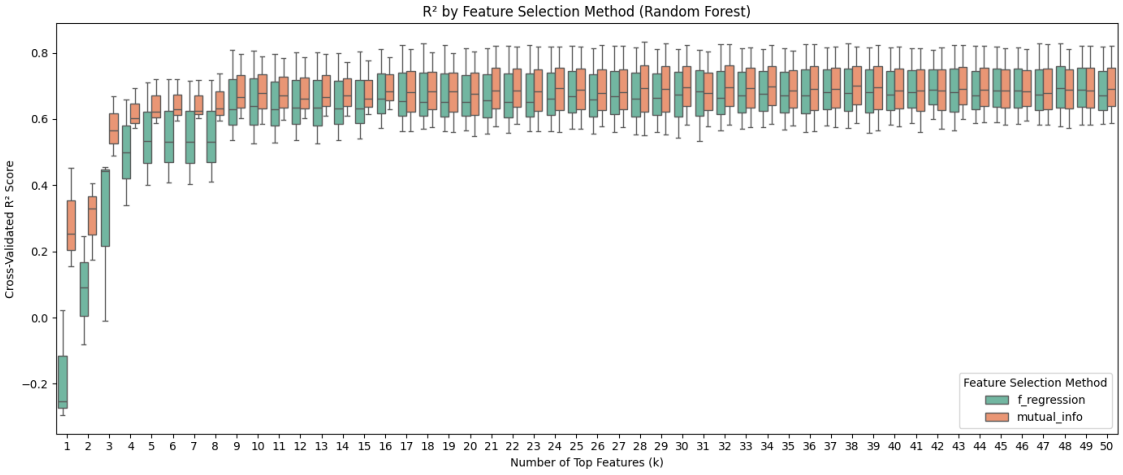


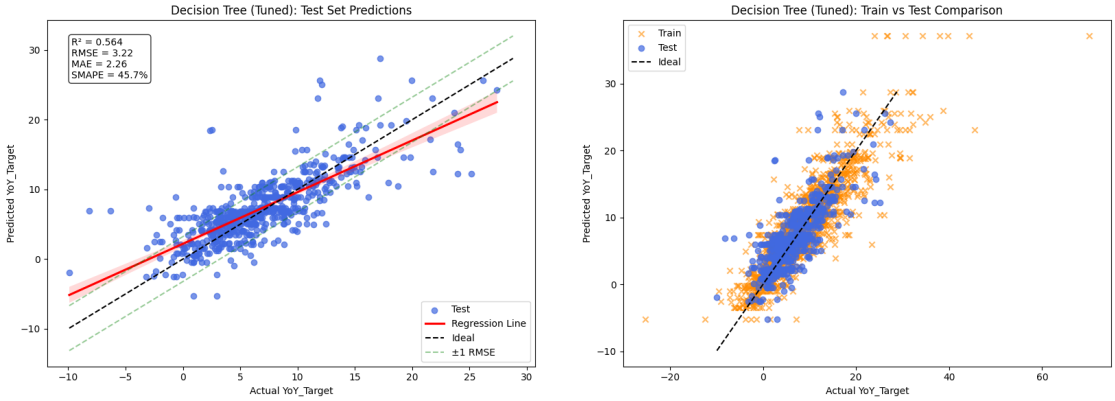
Figure 3.3: Feature selection performance for Random Forest

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

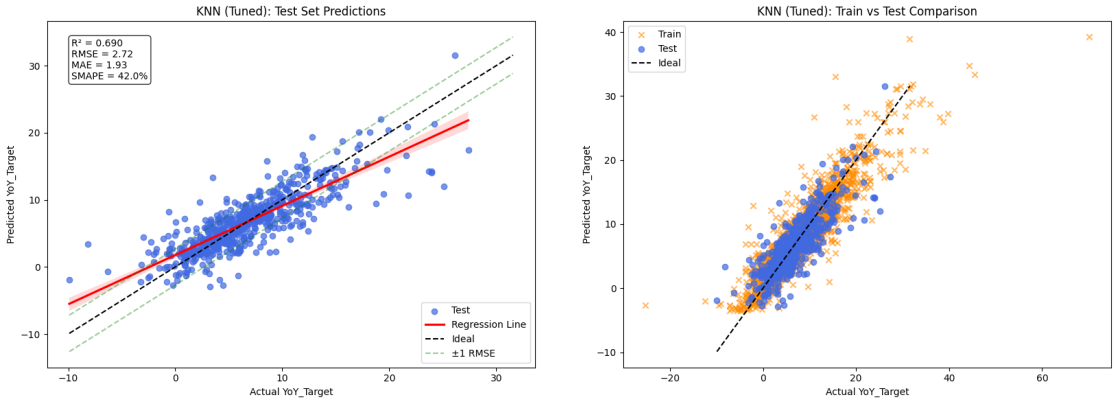


(a) (b)

Figure 3.4: Decision Tree test performance

The Decision Tree model underperforms relative to the other models. In Figure 3.4a, test predictions are concentrated near the mean, failing to capture the spread in actual values. The regression line is shallow and diverges significantly from the ideal diagonal. Figure 3.4b shows that while the model fits the training set tightly, it generalizes poorly. This indicates overfitting, which is consistent with its lower R^2 score and wider error margins on the test set.

3.2.2 K-Nearest Neighbors (KNN)



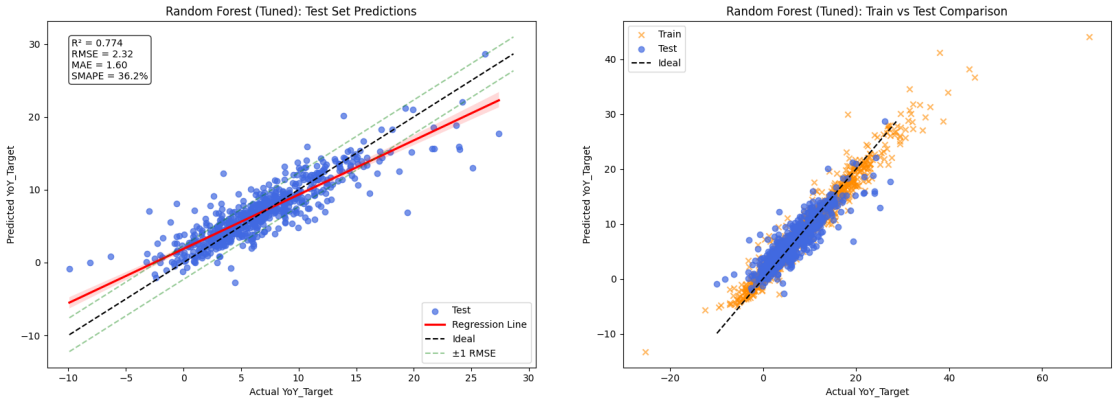
(a) (b)

Figure 3.5: KNN test performance

Figure 3.5a shows that KNN predictions align more closely with actual values compared to Decision Tree, especially in the mid-range. However, the model tends to underpredict at the higher end of the target range. In Figure 3.5b, training predictions are tighter than the test predictions, which show more scatter. This reflects the model's reliance on similar training

examples; when ZIPs with comparable patterns exist in the training data, KNN performs well, but it generalizes less effectively for outlier regions.

3.2.3 Random Forest



(a) (b)

Figure 3.6: Random Forest test performance

Figure 3.6a shows that Random Forest predictions are tightly clustered along the ideal $y = x$ line, with a regression slope closely matching the true trend. Outliers are limited, and the RMSE bands remain narrow. The train vs test comparison in Figure 3.6b reveals minimal difference between the two sets, demonstrating strong generalization. This model clearly captures both typical and atypical ZIP behavior more effectively than KNN or Decision Tree.

Table 3.1: Test set performance summary

Model	R^2	RMSE	MAE	SMAPE
Random Forest (16 features)	0.774	2.32	1.60	36.1%
KNN (9 features)	0.739	2.49	1.75	39.8%
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.

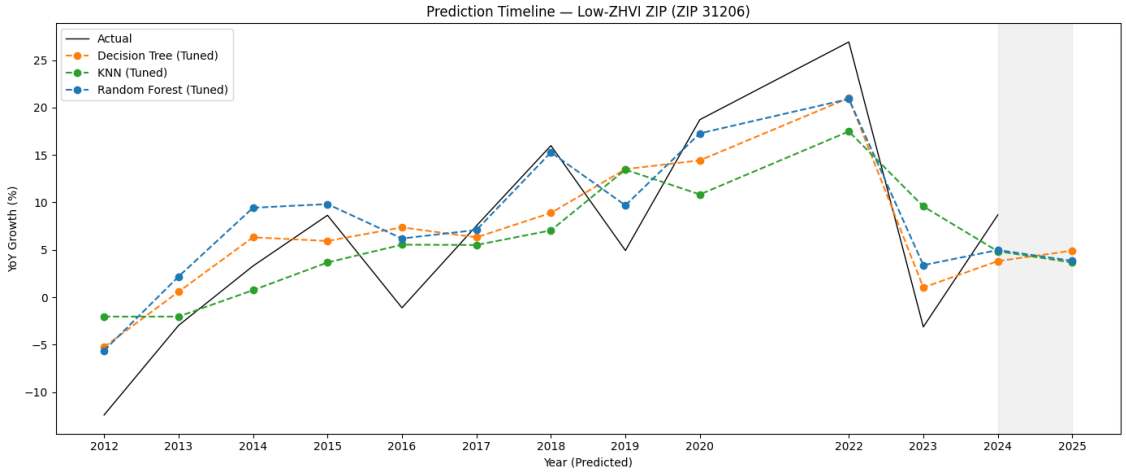


Figure 3.7: Prediction trends over time for low-ZHVI ZIP

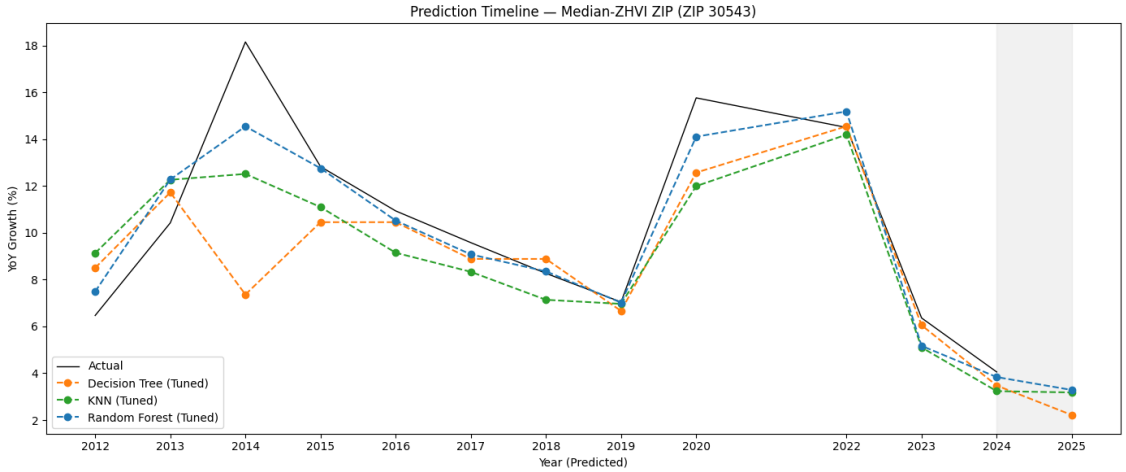


Figure 3.8: Prediction trends over time for median-ZHVI ZIP

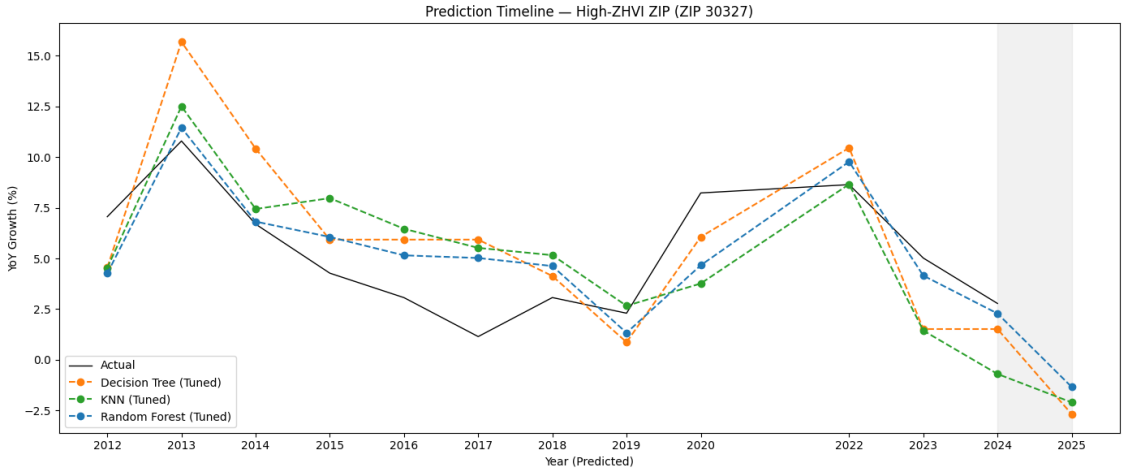


Figure 3.9: Prediction trends over time for high-ZHVI ZIP

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 and Analysis

Depending on the type of project you are doing, this chapter can be merged with "Results" Chapter as " Results and Discussion" as suggested by your supervisor.

In the case of software development and the standalone applications, describe the significance of the obtained results/performance of the system.

4.1 A section

The Discussion and Analysis chapter evaluates and analyses the results. It interprets the obtained results.

4.2 Significance of the findings

In this chapter, you should also try to discuss the significance of the results and key findings, in order to enhance the reader's understanding of the investigated problem

4.3 Limitations

Discuss the key limitations and potential implications or improvements of the findings.

4.4 Summary

Write a summary of this chapter.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This project explored the feasibility of modeling and predicting year-over-year growth in the Zillow Home Value Index (ZHVI) using historical housing and demographic features at the ZIP code level. The motivation stemmed from the need for data-driven insights to support homebuyers, investors, city planners, and policymakers in understanding local housing market dynamics and identifying high-value growth areas.

A robust pipeline was developed, starting from data collection (Zillow ZHVI and U.S. Census ACS data), followed by preprocessing, feature engineering, model training, and evaluation. Techniques like one-hot encoding, normalization, and feature selection using `SelectKBest` were critical in handling the complexity and high dimensionality of the data. Models including Decision Tree, K-Nearest Neighbors (KNN), and Random Forest were trained, tuned, and evaluated.

Among these, the **Random Forest model consistently outperformed** others, highlighting its ability to handle noisy and non-linear housing market data effectively. The ZIP-based data split helped prevent data leakage and ensured the generalizability of the model.

While the model achieved reasonable predictive performance, the limitations in data coverage and geographic scope highlighted areas for future improvement. This project demonstrates that machine learning models, when paired with proper data handling techniques, can serve as valuable tools for interpreting real estate trends and guiding informed decision-making.

5.2 Future Work

1. **Incorporating Material Price Data:** A significant extension would be integrating average material prices into the feature set. This would allow the model to better capture macroeconomic influences on housing costs, offering deeper insight into supply-side drivers of home value changes.

2. **Automated Data Updating with AI:** Leveraging AI to automate data ingestion from APIs (such as Zillow, Census, or commodity price feeds) can ensure that the dataset remains up-to-date without manual intervention. This will also allow the model to adapt dynamically as new information becomes available.

3. **Multi-Year Growth Modeling:** The current approach focuses on year-over-year changes. Extending this to multi-year trend forecasting would allow stakeholders to plan further into the future and could improve model robustness by capturing long-term dependencies.

4. **Improving Temporal and Geographic Flexibility:** Expanding the model to work across broader time periods and more geographic regions can increase its utility. This may involve integrating more granular spatial data or applying transfer learning approaches.

5. **Enhanced Visualization and Deployment:** Building a user-facing dashboard that visualizes ZIP code growth predictions and integrates filters for demographics, economic indicators, and material costs could greatly improve accessibility and real-world use.

References