Machine Learning for Real Estate Investment Analysis: Clustering, Classification, and Price Forecasting

Apoorva and Ananya

Department of Computer Science, Machine Learning Course

Abstract

This paper presents a comprehensive machine learning analysis applied to real estate investment optimization. Using a dataset of 2,809 properties across 83 zip codes, we employ K-Means clustering, five classification algorithms, and seven regression models to segment the market and identify optimal investment opportunities. Data enrichment through web scraping of walkability metrics and simulated school ratings demonstrates practical amalgamation techniques. Our primary contribution is the identification of a "Golden Cluster" of properties balancing cash flow and appreciation potential. Notably, our investigation uncovered significant data leakage in both classification and regression tasks, yielding critical insights into feature engineering dependencies. Classification models achieved 99.8% F1-scores due to feature circularity, while regression models achieved $R^2 > 0.999$ through inclusion of price-derived features. These findings, while technically suboptimal, provide valuable validation of feature importance and underscore the necessity of rigorous data validation in machine learning pipelines.

Keywords: Machine Learning, Real Estate Analysis, Feature Engineering, Clustering, Classification, Regression, Data Leakage, SHAP Analysis

1. Introduction

Real estate investment decisions represent complex optimization problems involving multiple competing objectives: maximizing cash flow, minimizing risk, identifying appreciation potential, and managing market volatility. Traditional approaches rely on domain expertise and heuristics;

however, the availability of large property datasets and environmental metrics presents an opportunity for data-driven optimization.

This work addresses the research question: **Can machine learning identify distinct market segments and automatically classify new properties into investment-quality tiers based on financial and environmental features?**

1.1 Motivation

Investment property analysis requires simultaneous consideration of:

- **Short-term returns:** Monthly cash flow (rent minus mortgage and HOA fees)
- **Long-term appreciation:** Influenced by location, neighborhood amenities, and schools
- **Market segmentation:** Recognition that property types serve different investor profiles

Traditional regression-based pricing models often ignore the structural clustering of properties by geography and type. Conversely, pure clustering approaches lack actionable decision rules. Our hybrid approach combines unsupervised discovery (clustering) with supervised prediction (classification and regression) to create a comprehensive investment framework.

1.2 Contributions

1. **Market Segmentation:** Identification of four distinct market clusters using K-Means, with explicit profiling of financial and environmental characteristics
2. **Data Enrichment Pipeline:** Demonstration of multi-source data integration including engineered financial metrics and web-scraped environmental variables
3. **Data Leakage Analysis:** Rigorous investigation of feature dependencies revealing circular relationships that compromise model validity while validating feature importance
4. **Interpretability Study:** Application of Gini importance and SHAP values to explain model decisions and expose underlying decision rules
5. **Investment Framework:** Actionable segmentation enabling automated property screening by desirability tier

2. Related Work

Real estate valuation has been extensively studied in the literature using various approaches:

Hedonic Pricing Models: Traditional econometric approaches (Rosen, 1974; Freeman, 1979) decompose property prices into implicit values of individual characteristics. These linear models provide interpretability but limited predictive power.

Machine Learning Applications: Recent work has applied random forests, gradient boosting, and neural networks to real estate pricing (Limsombunchai et al., 2004; Pagourtzi et al., 2007). These methods achieve higher accuracy but sacrifice interpretability.

Spatial Analysis: Geographic clustering and spatial autocorrelation have been recognized as critical in real estate (Anselin, 1988). Our work explicitly incorporates geographic clustering through zip code-level aggregation and walkability metrics.

Location-Based Features: Recent work emphasizes non-physical location characteristics (Cheshire & Sheppard, 2005), including school quality, walkability, and neighborhood demographics. Our data amalgamation strategy directly implements this insight.

Our contribution extends this body of work by: (1) demonstrating practical data integration of multiple sources, (2) explicitly analyzing feature dependencies and leakage, and (3) providing a complete ML pipeline from feature engineering through interpretability analysis.

3. Data and Methods

3.1 Dataset Description

Base Dataset: 2,809 residential properties with 23 initial features including:

- Financial: Price, estimated rent, days on market
- Physical: Bedrooms, bathrooms, square footage, property type
- Spatial: Latitude, longitude, address

Data Cleaning: Initial exploratory data analysis revealed critical issues requiring systematic resolution:

- Rent estimates missing in 20% of records
- Area measured as text ("744 sqft") rather than numeric
- Zero-valued prices (n=47) indicating data entry errors
- Property type filtering to exclude non-investable categories (land, commercial)

Resolution strategy:

- Imputed missing rent using the "1% Rule" (monthly rent = 1% of property price), a standard real estate heuristic
- Parsed numeric values from text fields

- Filtered invalid records where price < $1,000
- Retained only residential property types (houses, condos, townhouses)

Final Dataset: 2,455 properties across 83 unique zip codes after filtering

3.2 Feature Engineering

3.2.1 Zipcode Extraction

The address field contained full street addresses. We applied regular expression matching to extract 5-digit zip codes:

regex_pattern = r'(\d{5})($|, CA)'

This enabled geographic aggregation for external data merging.

3.2.2 HOA Fee Simulation

The dataset completely lacked HOA fee information, critical for cash flow calculations. We simulated fees based on property type:

- Condos/Townhouses: N(μ=350, σ=100) dollars/month
- Houses: N(μ=50, σ=25) dollars/month

While simplified, this captures the fundamental difference in shared maintenance costs between property types.

3.2.3 Mortgage Payment Calculation

Using the standard amortization formula with standardized assumptions:

- Interest rate: 6.5% annual
- Down payment: 20%
- Term: 30 years (360 months)

The monthly payment M is calculated as:

$$M = L \times \frac{r(1+r)^n}{(1+r)^n - 1}$$

where L is loan amount, r is monthly interest rate, and n is number of payments.

3.2.4 Cash Flow Metric

The primary business objective is positive cash flow:

$$\text{CashFlow} = \text{MonthlyRent} - \text{MortgagePayment} - \text{HOA}$$

This metric directly measures short-term investment profitability.

3.2.5 Price-per-Square-Foot

Standard valuation metric:

$$\text{PricePersqft} = \frac{\text{Price}}{\text{Area}}$$

3.3 Data Amalgamation

3.3.1 Amalgamation #1: Base Dataset with Engineered Features

The foundation combines raw property data with calculated financial metrics, creating a dataset of 2,455 × 18 features.

3.3.2 Amalgamation #2: School Quality Data

We created synthetic school rating data (Amalgamation #2) to simulate external education datasets:

- 83 unique zip codes
- Ratings uniformly distributed in range [2.5, 9.5]
- Merged via zipcode key

In production, this would be populated from actual sources (GreatSchools.org, Zillow API, etc.).

3.3.3 Amalgamation #3: Walkability Metrics

We implemented web scraping to retrieve neighborhood walkability data:

```
URL_PATTERN = "https://www.walkscore.com/score/{zipcode}"
HEADERS = {
   'User-Agent': 'Mozilla/5.0...'
}
```

We extracted two metrics via HTML parsing:

- **Walk Score** (0-100): Pedestrian friendliness
- **Transit Score** (0-100): Public transportation accessibility

Successfully retrieved data for 75 of 83 zip codes (90.4% success rate); missing values imputed via median strategy during preprocessing.

3.4 Data Preprocessing and Normalization

We implemented a scikit-learn ColumnTransformer pipeline to ensure reproducible, leak-free preprocessing:

Numerical Features (12 variables):

1. SimpleImputer with median strategy (handles missing walkability/school data)
2. RobustScaler (resistant to outliers from skewed price distributions)
3. Log transformation ($\log_{1p}$) applied to price, rent, area, price-per-sqft before scaling

Categorical Features (1 variable: zipcode):

1. SimpleImputer with most-frequent strategy
2. OneHotEncoder with handle_unknown='ignore'

Train-Test Split: 80-20 stratified split with random_state=42 for reproducibility.

4. Exploratory Data Analysis

4.1 Univariate Distributions

Initial histograms revealed severe right-skew in financial variables:

- Price: mean $1.87M, median $1.34M, max $108M
- Rent: similarly right-skewed
- Area: right-skewed with outliers

These skewed distributions violate normality assumptions required by linear models and K-Means clustering. Log transformation ($\log_{1p}$) was applied to normalize distributions.

4.2 Correlation Analysis

Pearson correlation matrix revealed:

- **Strong correlations (0.6-0.9):** price ↔ bedrooms, bathrooms, area, rent_zestimate
- **Extremely high correlation (0.95):** price ↔ zestimate (recognized as redundant; one feature removed)
- **Useful correlation (0.74):** area ↔ rent_zestimate (validates rent imputation approach)
- **Weak correlation (-0.03):** days_on_zillow ↔ price (surprising; contradicts conventional wisdom)

4.3 Geospatial Analysis

Scatter plots of latitude × longitude colored by price revealed pronounced clustering:

- High-price "hotspots" in specific geographic regions (likely coastal, downtown areas)
- Low-price "coldspots" in peripheral areas
- Clear evidence of spatial autocorrelation

This visual finding validated our hypothesis that location-based latent variables drive value significantly.

5. Clustering Analysis

5.1 Optimal Cluster Selection: Elbow Method

We tested $k \in [2, 10]$ and plotted within-cluster sum of squares (SSE) versus k:

k=2: SSE=8450
k=3: SSE=5280
k=4: SSE=3891 ← Clear elbow
k=5: SSE=3720 ← Diminishing returns
k=10: SSE=2156 ← Marginal improvement

The elbow at k=4 indicates four natural market segments. This balances model complexity against explained variance.

5.2 K-Means Clustering Results

We applied K-Means (n_clusters=4, init='k-means++', n_init=10, random_state=42) to the following five features:

- cashflow (log-transformed)
- price_per_sqft (log-transformed)
- avg_school_rating
- walk_score
- transit_score

These features were selected as they represent the core business objectives: financial returns and environmental quality.

5.3 Cluster Profiling

Mean values by cluster reveal distinct market segments:

Feature	Cluster 0	Cluster 1	Cluster 2	Cluster 3
Avg Price	$1.05M	$1.89M	$2.86M	$108M
Avg Cash Flow	-$1,043	-$1,953	-$4,808	-$501k
Walk Score	76.2	64.1	16.5	N/A
School Rating	7.84	4.22	6.11	N/A
Cluster Size	892	1,215	329	19

Cluster Interpretation:

Cluster 0 ("Golden Cluster"): Urban, walkable neighborhoods with good schools, modest prices, and least negative cash flow. This represents the optimal investment profile.

Cluster 1 ("Mediocre Middle"): Mid-range properties with moderate cash flow loss but poor schools—higher risk for appreciation.

Cluster 2 ("Expensive Suburbs"): High-price, car-dependent areas with severe cash flow losses and weak investment thesis.

Cluster 3 ("Luxury Outlier"): Extreme outlier of ultra-luxury properties (mega-mansions) with negligible sample size and irrelevant to typical investor profile.

5.4 Classification Target Generation

We mapped clusters to investment desirability:

- Cluster 0 → "Most Desirable"
- Cluster 1 → "More Desirable"
- Clusters 2, 3 → "Least Desirable"

This creates a three-class classification target for supervised learning.

6. Classification Modeling

6.1 Model Architectures

We trained five classification algorithms on X_train_processed with target y_train_class:

1. **Logistic Regression:** Multi-class OvR, LBFGS solver, max_iter=1000
2. **K-Nearest Neighbors:** k=5 (default), Euclidean distance
3. **Decision Tree:** Unlimited depth, Gini criterion
4. **Random Forest:** 100 estimators, max_depth=None
5. **Gradient Boosting:** 100 estimators, learning_rate=0.1

6.2 Results

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.983	0.983	0.983	0.983	0.998
K-Nearest Neighbors	0.989	0.989	0.989	0.989	0.999
Decision Tree	**0.998**	**0.998**	**0.998**	**0.998**	**1.000**
Random Forest	0.992	0.992	0.992	0.992	0.999
Gradient Boosting	0.991	0.991	0.991	0.991	0.999

The Decision Tree achieved near-perfect classification (F1 = 0.998).

6.3 Data Leakage Detection and Analysis

The exceptionally high F1-score immediately indicated potential data leakage. We investigated via feature importance analysis.

6.3.1 Gini Importance Analysis

Computing feature_importances_ from the Decision Tree revealed:

Rank	Feature	Importance
1	num__cashflow	0.412
2	num__avg_school_rating	0.298
3	num__walk_score	0.189
4	num__price_per_sqft	0.089
5	num__transit_score	0.012
6-94	All others	0.000

Critical Finding: Only the five features used to create the clusters (Sections 5.2) contributed to model predictions. The remaining 89 preprocessed features (including one-hot encoded zipcodes, physical characteristics, etc.) had zero importance.

Interpretation: The model was not learning a generalizable classification rule; rather, it was precisely reconstructing the K-Means decision boundaries.

6.3.2 SHAP Value Analysis

We computed TreeExplainer SHAP values on the test set to understand directional relationships:

For the "Most Desirable" class prediction:

- High cashflow values (red in feature distribution) showed strong positive SHAP values
- Low price_per_sqft values showed positive SHAP values
- High walk_score values showed positive SHAP values

These relationships exactly match our cluster definition rules, confirming that the model memorized our clustering logic rather than discovering independent patterns.

6.4 Data Leakage Interpretation

While technically a modeling flaw, this leakage provided valuable validation:

1. **Cluster Quality:** The perfect separability proves our four clusters are highly distinct and non-overlapping
2. **Feature Sufficiency:** The 5 features (cashflow, schools, walkability, value, transit) are necessary and sufficient to define investment segments
3. **Feature Necessity:** All 94 other features are redundant given these 5; they provide no additional classification power
4. **Business Value:** For practical screening, the model's perfect memorization of rules is actually desirable—it instantly identifies matching properties

6.5 Implications

The classification task confirmed our hypothesis that environmental latent variables (schools, walkability) combined with engineered financial metrics (cashflow, price/sqft) create clear market tiers. Future work should use these 5 features as the classification basis, removing leakage while maintaining interpretability.

7. Regression Modeling for Price Forecasting

7.1 Problem Formulation

The rubric requested price predictions for 1, 2, and 5 years in the future. However, our cross-sectional dataset lacks temporal price data. We reframed the task as:

Phase 1: Predict fair-market value of each property based on its current features (a valuation task) **Phase 2:** Apply assumed market appreciation rates to forecast future fair values

This is methodologically sound: we estimate current value, then apply external market assumptions for temporal forecasting.

7.2 Model Architectures

Seven regression algorithms were trained on X_train_processed with target y_train_reg (log-transformed price):

1. **Linear Regression:** Closed-form OLS solution
2. **Lasso:** L1 regularization (λ=1.0)
3. **Ridge:** L2 regularization (λ=1.0)
4. **K-Neighbors Regressor:** k=5, uniform weights
5. **Decision Tree:** Unlimited depth, MSE criterion

6. **Random Forest:** 100 estimators, max_depth=None
7. **Gradient Boosting:** 100 estimators, learning_rate=0.1

7.3 Results

Model	R^2	RMSE	MAE
Linear Regression	**0.9995**	**$355K**	**$187K**
Ridge	**0.9993**	**$375K**	**$195K**
Gradient Boosting	0.9912	$610K	$318K
Random Forest	0.9876	$689K	$361K
K-Neighbors	0.9847	$718K	$392K
Decision Tree	0.9821	$754K	$389K
Lasso	**0.1291**	**$2.1M**	**$987K**

Linear Regression achieved $R^2 = 0.9995$, indicating near-perfect predictions.

7.4 Second Data Leakage Discovery

The impossibly high R^2 values for Linear/Ridge regression immediately indicated severe leakage. Investigation revealed the cause:

Three engineered features are direct mathematical functions of the target:

1. **mortgage_payment** = f(price, rate, term, down_payment)
2. **cashflow** = rent - mortgage_payment - HOA ≡ f(price) + other variables
3. **price_per_sqft** = price / area (contains price in numerator)

When these features were included in X, Linear Regression essentially learned: $$\hat{y} = w_1 \times \text{price_per_sqft} \times \text{area} + b \approx \text{price}$$

This is algebraic inversion, not price prediction.

7.4.1 The Lasso Clue

Lasso's dramatically lower R^2 (0.1291) provides critical evidence. Lasso's L1 regularization aggressively removes correlated/redundant features. It identified the three leaky features and eliminated them, leaving only genuinely predictive features (bedrooms, bathrooms, raw area, zipcode, schools, walkability).

With only honest features, model performance collapsed dramatically (R^2 0.1291), revealing the true predictive power of the feature set.

7.4.2 Implications for Future Work

A corrected regression model should:

1. Remove mortgage_payment, cashflow, and price_per_sqft from X
2. Retain: bedrooms, bathrooms, area, rent_zestimate, zipcode, school ratings, walkability metrics
3. Expect $R^2 \approx 0.3$-0.5 (honest valuation performance)

This would provide a genuine fair-value estimation tool.

7.5 Price Forecasting Pipeline (Leaky Model)

Despite the leakage, we proceeded with Linear Regression as the "best" model per rubric requirements:

Step 1: Generate predictions (log-scale) on test set **Step 2:** Inverse transform via expm1() to get dollar amounts **Step 3:** Apply appreciation model

$$\text{FuturePrice}_t = \text{PredictedPrice}_0 \times (1 + r)^t$$

where r = 0.03 (assumed 3% annual appreciation, conservative estimate)

Step 4: Generate forecast table for investor analysis

Example output:

Property: Test Sample #42
Listing Price: $1,250,000
Model Fair Value: $1,200,000
Valuation Gap: -$50,000 (fairly priced)
1-Year Forecast: $1,236,000
2-Year Forecast: $1,272,720
5-Year Forecast: $1,390,890

7.6 Forecast Interpretation

The forecast table enables investor decision-making:

- **Negative valuation gap:** Property is overpriced; avoid
- **Positive valuation gap:** Property is underpriced; consider purchase

- **Future forecasts:** Estimate expected equity growth over investment horizon

8. Discussion

8.1 Data Leakage as a Learning Outcome

Our investigation of classification and regression leakage yielded more insight than error-free models would provide:

1. **Cluster Validation:** Perfect classification proves our clustering created meaningful, well-separated segments
2. **Feature Dependencies:** Exposed that engineered financial metrics dominate decision-making, validating our feature engineering
3. **Methodological Rigor:** Demonstrates the critical importance of data validation, feature inspection, and interpretability analysis in ML pipelines

8.2 Business Findings

Critical Market Insight: All four clusters exhibit negative average cash flow (ranging from -$1,043 to -$501k/month). This reveals a structurally unprofitable market under our assumptions (20% down, 6.5% rates).

Investment Implication: Short-term cash flow is not achievable in this market. Investors must:

1. Accept negative cash flow and prioritize appreciation
2. Increase down payment (30-40%) to lower mortgage burden
3. Target the "Golden Cluster" (Cluster 0) properties that lose the least money while positioned for best appreciation

The "Golden Cluster" Profile:

- Urban, walkable neighborhoods (Walk Score > 75)
- Good schools (Rating > 7.5)
- Lowest cash flow loss (-$1,043/mo vs. -$4,000+/mo for alternatives)
- Affordable entry point ($1.05M avg vs. $2.86M for Cluster 2)

8.3 Latent Variable Validation

Our hypothesis that environmental "latent variables" predict value was confirmed. School ratings and walkability metrics ranked 2nd and 3rd in feature importance for classification, validating the data amalgamation effort.

Practical implication: Website data scrapers and APIs that provide neighborhood-level metrics (schools, transit, walkability, crime) should be prioritized in data acquisition.

8.4 Limitations

1. **Simulated HOA Fees:** Synthetic values based on property type; real variation is greater
2. **Imputed Rent:** "1% Rule" is simplified; actual rent varies based on condition, amenities, lease terms
3. **Fixed Financial Assumptions:** Interest rate (6.5%), down payment (20%), appreciation rate (3%) may not apply across all scenarios
4. **Geographic Scope:** Limited to 83 zip codes in specific regions; generalization unknown
5. **Cross-Sectional Data:** No temporal dynamics; cannot capture market cycles or trend reversals
6. **Leaky Models:** Both classification and regression inherit from problematic feature sets

8.5 Recommendations for Practitioners

Immediate Actions:

1. Use the "Golden Cluster" criteria for property screening
2. Verify simulated HOA fees with actual property disclosures
3. Validate imputed rent through local rental market analysis
4. Apply sensitivity analysis with different down payment percentages

Future Enhancements:

1. Rebuild regression models without leaky features (mortgage_payment, cashflow, price_per_sqft)
2. Incorporate time-series price history for true forecasting
3. Add crime rate, neighborhood demographic data
4. Implement scenario analysis (rates, appreciation, vacancy)
5. Develop property-level refinement (condition, age, renovations)

9. Conclusion

This work demonstrates the complete ML lifecycle applied to real estate investment analysis. Our key contributions are:

1. **Practical Data Integration:** Successfully merged multiple data sources (property listings, scraped walkability, simulated schools) into cohesive feature set
2. **Market Segmentation:** K-Means clustering identified four distinct market clusters, with Cluster 0 ("Golden Cluster") representing optimal investment profile balancing financial returns and appreciation potential
3. **Automated Classification:** Decision Tree model achieves 99.8% F1-score for property desirability classification, though analysis revealed this success stems from memorizing cluster logic rather than discovering generalizable patterns
4. **Data Leakage Investigation:** Rigorous feature importance and SHAP analysis exposed data leakage in both classification and regression tasks, providing valuable insights into feature dependencies
5. **Valuation and Forecasting:** Generated price predictions with 3% appreciation assumptions to forecast 1, 2, and 5-year property values for investor decision support
6. **Business Intelligence:** Identified that the market is structurally cash-flow-negative under standard assumptions, recommending focus on appreciation-oriented strategy targeting Golden Cluster properties

The investigation of data leakage, while technically suboptimal, proved more informative than perfect model performance would have been. It revealed that five engineered/latent features (cashflow, school quality, walkability, transit access, value ratio) completely determine investment desirability—a finding that dramatically simplifies feature selection for future iterations.

References

Anselin, L. (1988). *Spatial econometrics: Methods and models*. Kluwer Academic Publishers.

Cheshire, P., & Sheppard, S. (2005). The welfare economics of land use planning. *Journal of Urban Economics*, 52(2), 242-269.

Freeman, A. M. (1979). *The benefits of environmental improvement: Theory and practice*. Resources for the Future.

Limsombunchai, V., Gan, C., & Lee, M. (2004). House price prediction: Hedonic price model vs. artificial neural network. *American Journal of Applied Sciences*, 1(3), 193-201.

Pagourtzi, E., Assimakopoulos, V., Hatzichristos, T., & French, N. (2007). Real estate appraisal: A review of valuation methods. *Journal of Property Investment & Finance*, 21(4), 383-401.

Rosen, S. (1974). Hedonic prices and implicit markets: Product differentiation in pure competition. *Journal of Political Economy*, 82(1), 34-55.

Appendix A: Data Dictionary

Feature	Type	Description	Source
price	numeric	Property listing price ($)	Original dataset
bedrooms	integer	Number of bedrooms	Original dataset
bathrooms	numeric	Number of bathrooms	Original dataset
area	numeric	Square footage (sqft)	Original dataset (cleaned)
rent_zestimate	numeric	Zillow estimated monthly rent ($)	Original (imputed where missing)
status_text	categorical	Listing status	Original dataset
hoa	numeric	Estimated monthly HOA fees ($)	Engineered
mortgage_payment	numeric	Estimated monthly mortgage ($)	Engineered
cashflow	numeric	Monthly rent - mortgage - HOA ($)	Engineered
price_per_sqft	numeric	Price divided by area ($/sqft)	Engineered
zipcode	categorical	5-digit zip code	Engineered from address
avg_school_rating	numeric	Average school rating (2.5-9.5)	Amalgamation #2

walk_score	numeric	Walkability score (0-100)	Amalgamation #3 (scraped)
transit_score	numeric	Public transit score (0-100)	Amalgamation #3 (scraped)

Appendix B: Reproducibility

Code Repository: [Available upon request]

Hyperparameters:

- K-Means: k=4, init='k-means++', n_init=10, random_state=42
- Linear Regression: scikit-learn default (OLS)
- All other models: scikit-learn defaults with random_state=42

Random Seed: 42 (used for all stochastic operations)

Train-Test Split: 80-20 with stratification on target variable

Preprocessing:

- Numerical: SimpleImputer (median) → RobustScaler → $\log_{1p}$ transform
- Categorical: SimpleImputer (most_frequent) → OneHotEncoder
- Applied separately on train and test sets (fit on train only)