Savills Property Forecasting System

AI-Assisted Forecasting and Decision Support for Singapore Private Residential Markets

SUTD Capstone Project — Final Report · Savills Property Forecasting System

Ruiqin Yang, Ruida Sun, Zihan Zhou, Haihong Huang, Bingqi Zhang, Linus Koh Zhijie

Abstract

This project develops a one-quarter-ahead forecasting system for Singapore's private residential market, with separate models for the mass market (Outside Central Region), mid-tier (Rest of Central Region), and luxury (Core Central Region) segments. The final workflow has four active layers: L0 prepares the data and engineered features; L1 generates AI-assisted sentiment scores; L2 compares OLS, ARIMA, Random Forest, XGBoost, DNN, and LSTM across 28 rolling one-quarter-ahead windows; and L4 applies SHAP for explainability. In the final run, Random Forest is selected for the mass market, while XGBoost is selected for the mid-tier and luxury segments. AI sentiment has the lowest measured importance among the 15 retained features. The system uses time-aware validation and empirical uncertainty intervals, with small-sample instability in the deep models as a key limitation.

Keywords: property price forecasting; Singapore residential market; machine learning; time series analysis; SHAP explainability; AI sentiment analysis

Executive Summary

This project forecasts Singapore private residential prices separately for the mass market, mid-tier, and luxury segments (OCR, RCR, and CCR). We combine historical prices, macroeconomic variables, policy information, and AI-derived sentiment with statistical, machine-learning, and time-series models. The final outputs include point forecasts, directional signals, empirical uncertainty intervals, and model explanations presented through a dashboard and advisor interface.

At L2, we compare six forecasting approaches: OLS, ARIMA, Random Forest, XGBoost, a deep neural network (DNN), and LSTM. Each model is tested one-quarter-ahead across 28 rolling windows using 30- and 40-quarter training windows. We report RMSE, MAE, and directional accuracy, with ARIMA(1,1,0) and ARIMA(2,1,0) serving as classical time-series benchmarks.

The final 28-window backtest run on 2026-08-12 selects a tree model for each segment:

Table 1. Best model per segment (final run).

Segment	Best model	Window / params
Mass market	RandomForest	40 / n_estimators=100, max_depth=5
Mid-tier	XGBoost	30 / lr=0.05, max_depth=3
Luxury	XGBoost	30 / lr=0.1, max_depth=5

Source: data/outputs/all_parameters_tested_{mass_market,mid_tier,luxury}.csv; data/outputs/predictions.csv

Final one-quarter-ahead point forecasts with empirical residual-based prediction intervals:

Table 2. Final point forecasts with empirical prediction intervals.

Segment	Latest price	Forecast change	Forecast Q1	80% PI	95% PI	Direction accuracy
Mass market	$1,873.0	+2.31%	$1,916.84	$1,804–2,094	$1,722–2,198	85.7%
Mid-tier	$2,217.0	+1.82%	$2,257.79	$2,081–2,452	$1,940–2,837	67.9%
Luxury	$2,975.0	+0.00%	$2,975.00	$2,834–3,124	$2,797–3,280	85.7%

Source: data/outputs/predictions.csv (28 backtest windows, method L2 direct)

The 95% prediction intervals are wide; for example, the mass-market interval is roughly ±12%. This reflects the limited 28-window backtest, so the point forecasts should be interpreted together with their uncertainty ranges.

Performance differs across the three segments, which supports selecting models separately rather than imposing one model on the whole market. Random Forest is selected for the mass market, while XGBoost is selected for the mid-tier and luxury segments. The mass-market (+2.31%) and mid-tier (+1.82%) forecasts are modestly positive, whereas the luxury forecast is flat (+0.00%). AI-derived sentiment remains in the feature set but ranks last in the final SHAP analysis (mean |SHAP| = 0.0036), so we treat it as supporting information rather than a core driver.

The report uses time-aware rolling validation, empirical uncertainty estimates, and SHAP explainability, while clearly stating the limitations created by the small quarterly sample and the current implementation.

1. Introduction

1.1 Background and Motivation

Singapore residential property prices are shaped by macroeconomic conditions, financing costs, government cooling measures, supply, and market expectations. Traditional models capture historical relationships well, but qualitative policy and sentiment information is harder to represent directly. We therefore combine structured market data with statistical, machine-learning, time-series, and AI-assisted sentiment methods.

1.2 Problem Statement

Savills needs a decision-support approach that can forecast both market direction and price movement across distinct residential segments while remaining interpretable for professional use. We focus on three questions: which variables are most useful for each segment, which modelling approach generalises best through time, and how the forecasts can be communicated with clear uncertainty and explanatory context.

1.3 Objectives

- Build a reproducible quarterly feature pipeline for three Singapore private residential market segments.
- Compare statistical, machine-learning, and time-series forecasting approaches using time-ordered validation.
- Quantify directional performance and forecast error using multiple metrics.
- Incorporate policy and sentiment context without introducing data leakage.
- Provide uncertainty estimates, explainability and a usable dashboard/chatbot interface.

Section 4.7 links the main modelling choices to the literature reviewed by the team.

2. System Architecture

The final architecture has four active layers. L0 handles data and feature engineering, L1 produces the sentiment feature, L2 trains and evaluates the forecasting models, and L4 provides SHAP explainability and supports the user interface. The earlier L3 Bayesian fusion module remains in the codebase only as an experimental reference and is not used in the final forecasts; Section 6.2 explains this decision.

Table 3. Architecture layers and roles.

Layer	Role	Final report positioning
L0 Data	Savills prices + macro/policy/external features	Core data foundation
L1 AI sentiment	Quarterly qualitative score (3 analyst personas)	Supplementary explanatory feature
L2 Models	OLS, ARIMA, RF, XGBoost, DNN, LSTM	Core six-model forecasting comparison
L3 Bayesian fusion	Removed from active pipeline	Earlier experimental method only

Layer	Role	Final report positioning
L4 Explainability	SHAP feature importance	Interpretability and business insight
Interface	CLI, web dashboard, chatbot/advisor	Decision-support delivery layer

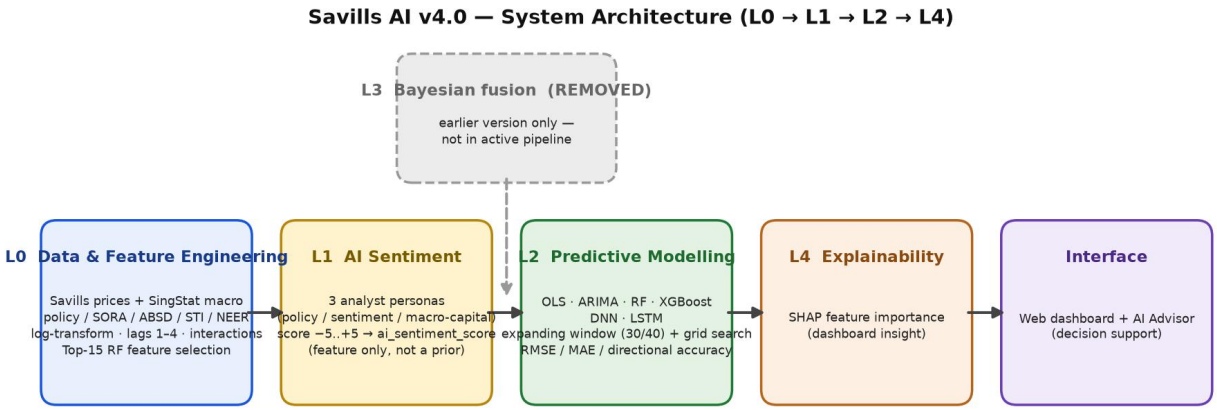


Figure 1 — Savills Property Forecasting System architecture: L0 data and feature engineering → L1 AI sentiment (feature only, not a prior) → L2 six-model forecasting comparison → L4 SHAP explainability, delivered through the dashboard and advisor interface. L3 Bayesian fusion is not part of the active pipeline (Section 6.2).

3. Data and Feature Engineering

3.1 Data Sources

We combine Savills segment-level property prices with Singapore macroeconomic indicators and external market features. The feature set includes nominal GDP, CPI, unemployment, SORA, ABSD change indicators, STI, SGD NEER, AI sentiment, and engineered lags and interactions. The external feature file contains 73 quarterly observations from 2008Q1 to 2026Q1. Private supply is included structurally but is missing for all 73 rows and is currently filled with 0.0 during training; this feature should either be populated and validated or removed.

Source: data/external_features.csv (73 rows; private_supply 0/73 non null); utils/config.py:51-77

3.2 Transformations and Feature Engineering

Property prices and nominal GDP are log-transformed. We generate one- to four-quarter lags for price and GDP, and include interaction terms such as CPI × nominal GDP and log GDP × AI sentiment. A Random Forest feature-selection stage then retains the top-15 features to reduce overfitting on the small sample.

Source: utils/config.py:82-86, 141; pipeline/data_processor.py

3.3 Data Quality and Leakage Controls

- Preserve chronological order; quarterly observations are never randomly shuffled during forecast evaluation.
- Fit feature selection, scaling, and model parameters only on the training portion of each rolling window.
- Track the publication and availability timing of macroeconomic variables when describing the system as a real-time forecasting tool.

The following sources, frequencies, and latest vintages are confirmed from the delivered code and data files:

- Nominal GDP — SingStat table M015651, quarterly; latest vintage 2026Q1; cached fallback in data/macro_fallback.csv (2008Q1–2026Q1).
- CPI (all items) — SingStat table M213751, monthly, aggregated to quarterly; latest vintage 2026 May.
- Unemployment rate — SingStat table M183401, annual frequency, interpolated to quarterly; the fetcher flags this granularity as insufficient.
- SORA (3M), ABSD change indicator, STI index and SGD NEER — supplied in data/external_features.csv (73 quarterly rows, 2008Q1–2026Q1) as external series inputs rather than SingStat pulls.
- Private supply (private_supply) — structurally present in external_features.csv but entirely unpopulated (0/73 rows); currently filled with 0.0.

Release lags are not yet recorded explicitly in the codebase. At run time, the L0 fetcher retrieves the latest available vintage from the SingStat Table Builder API. Recording the authoritative release calendar for each series remains a documentation task (Section 8).

- Resolve the missing private-supply series or remove the feature rather than treating missing observations as true zeros.
- Version the dataset used to generate the final report tables and figures.

4. Methodology

4.1 Forecasting Formulation

The primary task is one-quarter-ahead forecasting for each market segment. We use expanding rolling windows rather than random cross-validation, testing 30- and 40-quarter training windows and producing 28 one-quarter-ahead test points. The configuration file also defines training (2008Q1–2022Q4), validation (2023Q1–2024Q4), and blind-test periods (2025Q1–2026Q1). However, the current L2 grid-search implementation does not enforce these date

cutoffs; it expands from an initial 40-quarter training window. We therefore treat the fully separated blind test as future work.

Source: utils/config.py:133-138; models/ml_trainer.py (INITIAL_TRAIN_SIZE=40)

4.2 Models

The active L2 comparison covers six approaches: OLS, ARIMA, Random Forest, XGBoost, a deep neural network (DNN), and a long short-term memory network (LSTM). Together they represent four modelling families: linear econometric models, classical univariate time series, tree ensembles, and neural networks. Logistic Regression appears only in earlier exploratory classification work and is not part of the final forecasting comparison.

4.3 Time Series Component (ARIMA)

ARIMA provides the classical time-series benchmark. We compare ARIMA(1,1,0) and ARIMA(2,1,0) using 30- and 40-quarter rolling windows. First differencing (d = 1) addresses the unit-root behaviour of the price levels, while p = 1–2 keeps the specification parsimonious for quarterly data. At each test point, ARIMA is fitted on the preceding window, forecasts the next quarter, and then rolls forward. This benchmark tests whether recent price dynamics alone can compete with models that also use external variables.

Source: utils/config.py:126; models/ml_trainer.py

As a supplementary check, we computed stationarity diagnostics on 2026-08-16 using the 69-quarter log-price series in data/processed/feature_store.csv (2009Q1–2026Q1). These diagnostics support the ARIMA specification but were not part of the final backtest.

The Augmented Dickey-Fuller (ADF) test fails to reject a unit root in the log price level for all three segments: mass market ADF = −1.72 (p = 0.42), mid-tier ADF = −0.89 (p = 0.79), and luxury ADF = −0.61 (p = 0.87). After first differencing, all three series are stationary: mass market ADF = −5.40 (p < 0.001), mid-tier ADF = −5.69 (p < 0.001), and luxury ADF = −4.13 (p = 0.001). This supports d = 1 in ARIMA(1,1,0) and ARIMA(2,1,0).

Figures 2–4 show ACF and PACF diagnostics for each segment. The top row uses log price levels, while the bottom row uses first differences (quarter-on-quarter log changes). The shaded bands are 95% confidence bands; bars outside the bands indicate statistically significant autocorrelation.

- Level ACF (top left): slow decay across many lags, consistent with a non-stationary price level.
- Level PACF (top right): one dominant lag-1 spike followed by values close to zero, consistent with random-walk-like behaviour.
- Differenced ACF and PACF (bottom row): most bars return inside the confidence band within one or two lags, supporting the d = 1 specification.

- The limited autocorrelation after differencing also shows that quarter-on-quarter price changes are difficult to predict from price history alone.

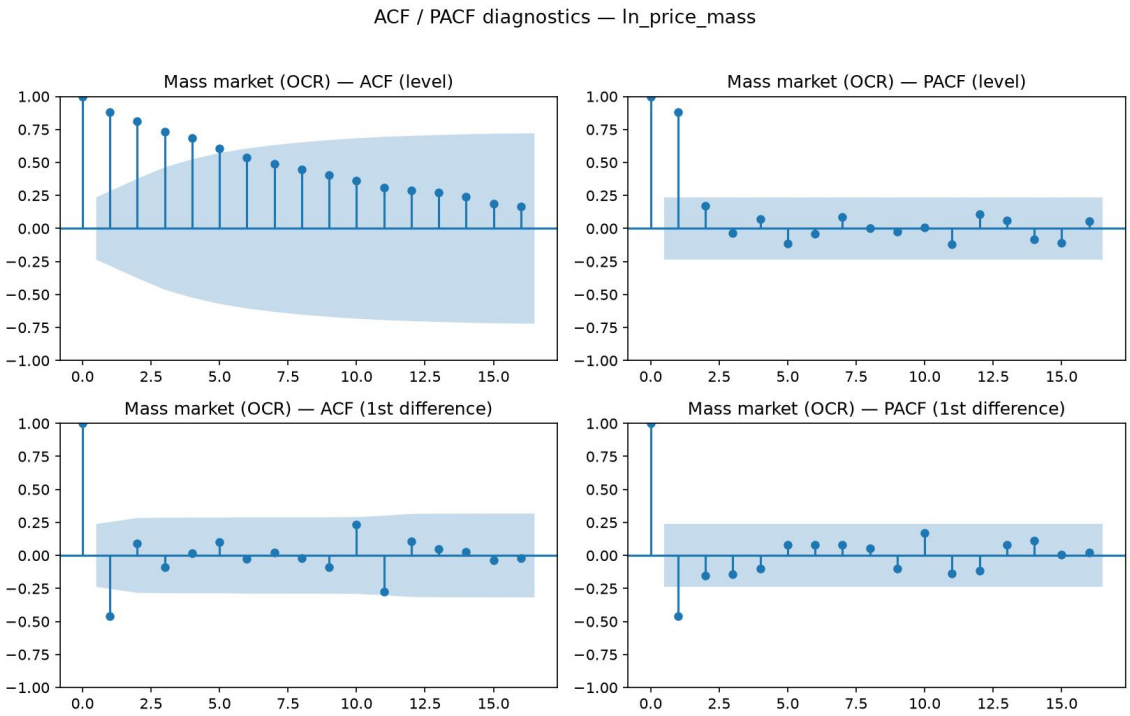


Figure 2 — ACF/PACF diagnostics for mass-market log price (level vs first difference).

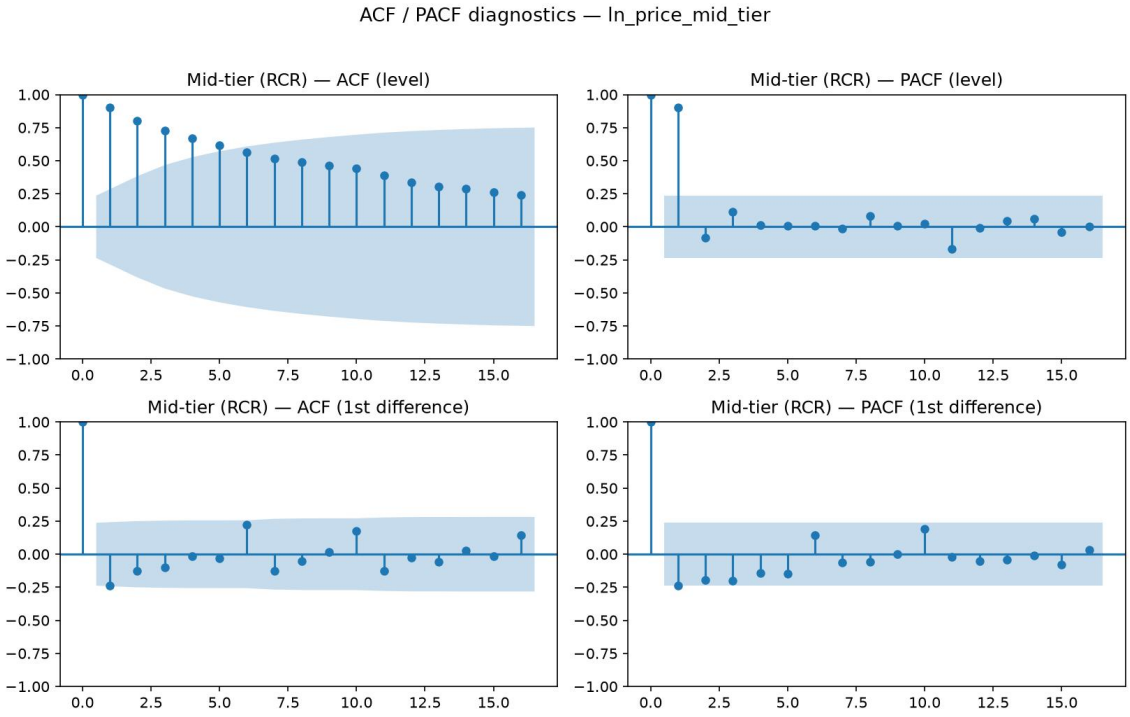


Figure 3 — ACF/PACF diagnostics for mid-tier log price (level vs first difference).

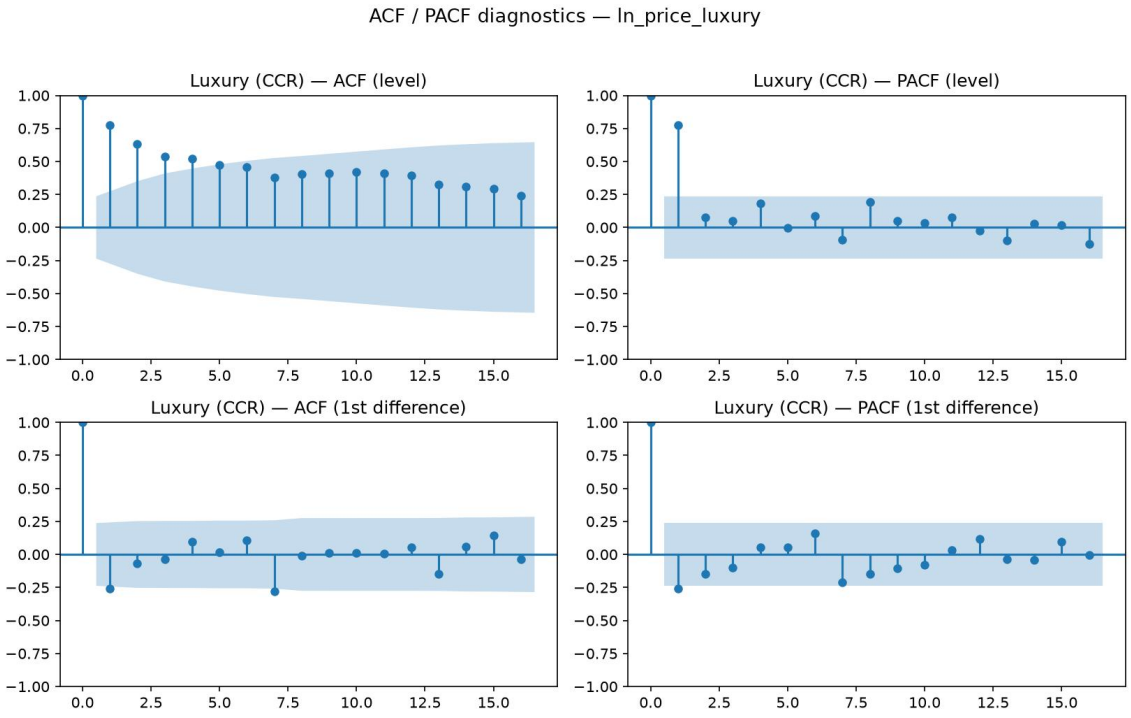


Figure 4 — ACF/PACF diagnostics for luxury log price (level vs first difference).

4.4 AI Sentiment and Neural Network Implementation

L1 converts policy, market-sentiment, and macro-capital information into a score from −5 to +5. The score enters L2 as a structured feature and is also intended for a GDP × sentiment interaction. Three fixed analyst roles assess each quarter independently: policy_compliance focuses on regulation, ABSD, and LTV; market_sentiment focuses on buyer behaviour and demand; and macro_capital focuses on capital flows, exchange rates, and interest rates. Each role returns a JSON score with a short rationale, and the three scores are averaged equally. If the live language model is unavailable, the module uses a deterministic fallback and caches the result.

Source: utils/config.py:100-121; agents/sentiment_analyst.py

In the current LSTM implementation, each feature row is passed with sequence length 1. Most temporal information therefore comes from engineered lag variables rather than recurrent sequence learning. A later experiment should test genuine multi-quarter input sequences (Section 8).

4.5 Evaluation Metrics

We evaluate models with RMSE, MAE, and directional accuracy. RMSE penalises larger errors, MAE gives the average absolute error, and directional accuracy records whether the quarter-on-quarter move has the correct sign. Directional accuracy is reported as the headline business

metric because Savills prioritises the direction of the next move, while RMSE remains the primary model-selection metric in the current implementation.

4.6 Model Selection Strategy

Model selection is performed separately for each segment. In the final implementation, candidate configurations are compared across the rolling windows using RMSE as the primary metric, with MAE and directional accuracy as supporting measures. The best single model is then used for the L2-direct forecast. As noted in Section 4.1, the configured blind-test date cutoffs are not currently enforced, so the reported results should not be interpreted as evaluation on a fully untouched holdout.

Source: app.py:88-220

4.7 Literature Review and Design Rationale

This section links the modelling choices in Sections 4.1–4.6 to the literature reviewed by the team. The ten studies cover the main model families used in L2, the supporting methodological choices, and the decision to remove Bayesian fusion. Full references are listed in Section 10.

Literature landscape.

We reviewed ten empirical and methodological studies of Singapore's housing market across seven method families. The studies are grouped below according to their relevance to our modelling choices:

• **Hedonic / linear regression (OLS).** Xu (2017) constructs a resale HDB hedonic price index using OLS with time dummies, district fixed effects and 20-fold cross-validation (full model RMSE ≈ 2.12). Liu & Pan (2025) apply multiple linear regression with correlation and VIF analysis to select Singapore resale price determinants. Cao et al. (2019) and Wang, Cai et al. (2022) retain OLS as the non spatial baseline against which spatial variants are compared. These studies support the use of OLS as the linear baseline in this project.

• **Classical time series (ARIMA).** Lim et al. (2016) benchmark ARIMA against an ANN on the condominium price index and find the ANN ahead (MAE/RMSE 1.9/3.1 vs 3.2/5.0). ARIMA therefore serves as the univariate benchmark for comparison with the multivariate and nonlinear models.

• **Neural networks.** Lim et al. (2016) and Wang, Chan et al. (2016) both find static and delayed/dynamic ANNs competitive for Singapore price index and resale prediction; Durai & Wang (2023) use a neural network downstream of sentiment features. Based on these studies, a neural network model was retained for comparison.

• **Tree ensembles / boosting.** Sing et al. (2022) report the best automated valuation results from boosted tree ensembles on more than 300,000 public and private housing transactions. Their private housing model explains 94.28% of price variance with a MAPE of 5.34%, providing relevant evidence for including tree ensembles in this study.

• **Spatial regression (GWR).** Cao et al. (2019) and Wang, Cai et al. (2022) show geographically weighted regression (Euclidean and travel time) outperforms OLS for explaining spatial price variation. This family is reviewed but not adopted: the forecasting task uses segment-level indices rather than address level cross sections, so there is no location distance feature to weight spatially.

• **Structural / equilibrium models.** Huang & Zhang (2012) use a simultaneous equation (3SLS) model with structural change detection, and Chia et al. (2017) a dynamic general equilibrium model, to explain how fundamentals drive price formation. Neither reports point forecast error metrics; they motivate the macro fundamental feature set rather than the forecast models themselves.

• **NLP / sentiment.** Durai & Wang (2023) add VADER COVID-19 tweet sentiment (with Granger causality filtering) to a neural forecaster and report improved accuracy over structure only baselines — the direct precedent for the L1 AI sentiment feature.

Theoretical basis for each model in the L2 comparison.

The six forecasting approaches were included for the following reasons:

• **OLS — hedonic pricing and linear attribution.** Housing value is conventionally modelled as a linear function of observable attributes (hedonic pricing theory). OLS provides an interpretable baseline, while more flexible models are retained only when they improve out-of-sample performance. Precedent: Xu (2017); Liu & Pan (2025).

• **ARIMA — Box–Jenkins time series identification.** Price levels are non-stationary (unit root confirmed by the ADF/ACF diagnostics in §4.3), so first differencing (d = 1) is applied; p = 1–2 is parsimonious for quarterly data. ARIMA isolates whether autoregressive price dynamics alone can forecast next quarter movement. Precedent: Lim et al. (2016).

• **Random Forest — bagged decision trees.** Random Forest was included as a nonlinear alternative to OLS because it can represent interactions and threshold effects without specifying them in advance. It also provides a bagging-based comparison with the boosting approach discussed by Sing et al. (2022).

• **XGBoost — gradient boosted trees.** Gradient boosting sequentially fits weak trees to residual error and is the strongest single precedent for Singapore housing valuation (Sing et al., 2022). A tree ensemble is therefore used in the segments where it records the best validation result.

• **DNN — universal function approximation.** A feed-forward network can approximate arbitrary nonlinear mappings; Lim et al. (2016) and Wang, Chan et al. (2016) report strong ANN results on Singapore data. The network was retained for comparison with those studies, although it is unstable on the small quarterly sample of roughly 40 training observations (§5.4) and is not used for the final forecast.

• **LSTM — recurrent sequence modelling.** Recurrent architectures are designed for sequential dependence; the delayed/dynamic networks of Wang, Chan et al. (2016) motivate sequence-aware neural models. It is retained for comparison, but with sequence length 1 in the current implementation (§6.3) its temporal context comes from engineered lags rather than recurrence.

• **AI sentiment — qualitative signal as a regressor.** Durai & Wang (2023) show sentiment improves prediction over structural variables alone; the L1 score carries qualitative policy and sentiment information that lagged prices and macro fundamentals do not.

• **SHAP — Shapley value attribution.** SHAP computes Shapley values, the unique additive feature attribution satisfying the axioms of cooperative game theory (efficiency, symmetry, dummy, additivity). The resulting feature contributions provide a consistent way to examine how the model uses its inputs (this methodological reference is separate from the ten housing studies reviewed by the team).

• **Top 15 RF feature selection — overfitting control.** With ~40–68 training rows and ~29 engineered features, retaining all features risks overfitting; RF importance ranking retains the top-15. Precedent: Liu & Pan (2025) use correlation and VIF for the same purpose in a linear setting.

The literature also supports several implementation choices. We model OCR, RCR, and CCR separately because housing submarkets can have different price relationships and pooling them can reduce prediction accuracy (Goodman & Thibodeau, 2003). The final results also show different model performance by segment. We log-transform prices and GDP to stabilise variance and linearise multiplicative growth (Rosen, 1974), and generate one- to four-quarter lags because property prices show persistence and annual seasonality (Box & Jenkins, 1970). The CPI × GDP term captures the joint effect of inflation and nominal activity. The GDP × sentiment term was intended to vary the sentiment effect with economic conditions, but it is currently computed before sentiment injection and therefore requires revision (Section 6.3). Directional accuracy is the headline business metric because advisory decisions depend heavily on the sign of the next move, while point-error metrics are noisy on a 28-window sample (Hyndman & Athanasopoulos, 2021; Bergmeir & Benítez, 2012). The three analyst roles score policy, market, and capital-flow conditions from −5 to +5, with a structured JSON response storing both the numeric score and a short rationale (Wei et al., 2022). Observations remain in time order during evaluation (Bergmeir & Benítez, 2012), and cached sentiment plus a deterministic fallback keeps the pipeline reproducible when the API is unavailable (Peng, 2011).

Why the Bayesian fusion layer was removed.

We removed the earlier L3 Bayesian fusion layer for four methodological reasons. First, in v3.0 the same AI sentiment signal entered L2 as a feature and L3 as a prior, creating a risk of double counting. Second, the prototype used a fixed 50% prior weight while the MCMC backend was inactive, and no ablation test showed that this weighting improved forecasts. Third, none of the ten reviewed studies provided an external prior that could be transferred defensibly to our

setting. Finally, with a small quarterly sample, empirical intervals from rolling residuals require fewer assumptions than a posterior that depends heavily on an unvalidated prior. We therefore use direct L2 point forecasts with residual-based prediction intervals. The Bayesian module remains in the codebase only as an earlier experimental reference.

The main implementation choices can be traced to config.py, ml_trainer.py, data_processor.py, sentiment_analyst.py, shap_explainer.py, and the team's design note. The most important decisions are segment-specific model selection, directional accuracy as the headline business metric, and direct L2 forecasts with residual-based intervals instead of Bayesian fusion.

5. Results

5.1 Model Selection

Table 4. Winning configuration per segment used for the final forecast.

Segment	Best model	Window	Parameters
Mass market	RandomForest	40	n_estimators=100, max_depth=5
Mid-tier	XGBoost	30	lr=0.05, max_depth=3
Luxury	XGBoost	30	lr=0.1, max_depth=5

Source: data/outputs/predictions.csv

5.2 Final One-Quarter-Ahead Forecasts with Empirical Prediction Intervals

For each segment, the selected model is retrained on all available data and used to forecast the next quarter. The 80% and 95% intervals are derived from percentiles of log-price errors across the 28 rolling backtest windows, rather than from a Bayesian posterior.

Table 5. Final one-quarter-ahead forecasts with empirical prediction intervals.

Segment	Latest	Change	Forecast Q1	80% PI	95% PI	Direction accuracy
Mass market	$1,873.0	+2.31%	$1,916.84	$1,804–2,094	$1,722–2,198	85.7%
Mid-tier	$2,217.0	+1.82%	$2,257.79	$2,081–2,452	$1,940–2,837	67.9%
Luxury	$2,975.0	+0.00%	$2,975.00	$2,834–3,124	$2,797–3,280	85.7%

Source: data/outputs/predictions.csv (method L2 direct, 28 backtest windows)

The mass-market (+2.31%) and mid-tier (+1.82%) forecasts are modestly positive, while the luxury forecast is flat (+0.00%). The 95% prediction intervals remain wide; for example, the mid-tier range is $1,940–$2,837. These intervals reflect the small sample and should always be reported alongside the point forecasts.

5.3 Feature Importance and Explainability (SHAP)

In the final SHAP output, lagged property prices and nominal GDP make the largest contributions among the 15 retained features, with the CPI × GDP interaction also contributing. AI sentiment ranks 15th of 15 (mean |SHAP| = 0.0036, compared with 0.0258 for the top feature). The five highest-ranked features are:

Table 6. SHAP feature importance (top-15 retained features).

Rank	Feature	Mean |SHAP|
1	ln_price_mid_tier_lag2	0.0258
2	ln_price_mass_lag4	0.0188
3	ln_nominal_gdp_lag4	0.0137
4	ln_nominal_gdp_lag2	0.0116
5	cpi_x_gdp	0.0113
15	ai_sentiment_score	0.0036

Source: report_output/shap_importance.csv

AI sentiment has the lowest SHAP importance among the retained features, so we treat it as a supporting feature until an ablation study can test its out-of-sample value. An earlier RF-based feature-importance file (data/outputs/feature_importance.csv) is stale and includes target_forward itself as a feature, which is a leakage artifact; it should not be used. The current ml_trainer.py excludes the target from the feature set.

5.4 Model Stability

With training windows of roughly 40 observations, the deep models are unstable. The single-hidden-layer DNN produces much larger errors than the tree and OLS models, while LSTM directional accuracy is close to 50%, consistent with its sequence length of 1. We therefore use one selected model per segment rather than an ensemble that includes unstable deep models.

6. Discussion

6.1 Key Findings

- Market segmentation matters: the best model differs by segment (Random Forest for the mass market; XGBoost for the mid-tier and luxury segments).
- XGBoost and Random Forest give the strongest overall results, consistent with earlier housing studies that favour tree ensembles.
- The mid-tier segment is the hardest to predict directionally, contrary to the earlier baseline that suggested luxury was closest to random.
- Lagged prices are the most informative features, consistent with persistence and momentum in quarterly property prices.
- AI sentiment is the weakest of the 15 retained features in the final run; its out-of-sample value is not yet demonstrated.

6.2 Bayesian Fusion in Earlier Versions

Bayesian fusion is not part of the active forecasting pipeline. We removed it because the prototype did not demonstrate incremental out-of-sample value over the direct L2 forecasts. In addition, AI sentiment was counted as both a feature and a prior, and the fixed 50% prior weight was not empirically calibrated. A separate ensemble experiment was also sensitive to unstable DNN and LSTM predictions, producing a +10.29% mid-tier anomaly; this reinforced the decision not to add an unvalidated fusion layer on top of unstable components. The mid-tier directional result did not improve.

The final pipeline therefore uses the selected L2 model directly and derives uncertainty intervals from the rolling backtest error distribution (Section 5.2). The Bayesian code is retained only as an experimental reference.

Source: app.py:88-89; requirements.txt:28 (pymc commented out); v4.0_新方案_移除贝叶斯.md

6.3 Limitations

- The small quarterly sample (~40–68 observations) limits data-intensive models such as DNN and LSTM; the 28-window backtest is also statistically noisy.
- Macroeconomic data may be subject to revision and publication lags.
- The private-supply feature is entirely missing (0/73 rows) and is currently filled with 0.0; a missing value should not be interpreted as a true zero.
- The configured train/validation/blind-test dates are not enforced by the current L2 rolling implementation, so model selection may use observations intended for final evaluation.
- The GDP × sentiment interaction is computed before the sentiment score is injected, so it does not yet represent the intended interaction. The feature should be recomputed after sentiment injection; until then, it should be interpreted mainly as a GDP-driven term.
- The current LSTM uses sequence length 1, so most temporal information comes from engineered lags rather than recurrent sequence learning.
- DNN and LSTM are unstable or close to chance on this sample size; they are retained for comparison but not used in the final forecast.
- Luxury-market drivers may be underrepresented because foreign-capital and supply variables are limited or missing.
- AI sentiment ranks last in SHAP, and its out-of-sample contribution has not yet been established.

7. Business Application

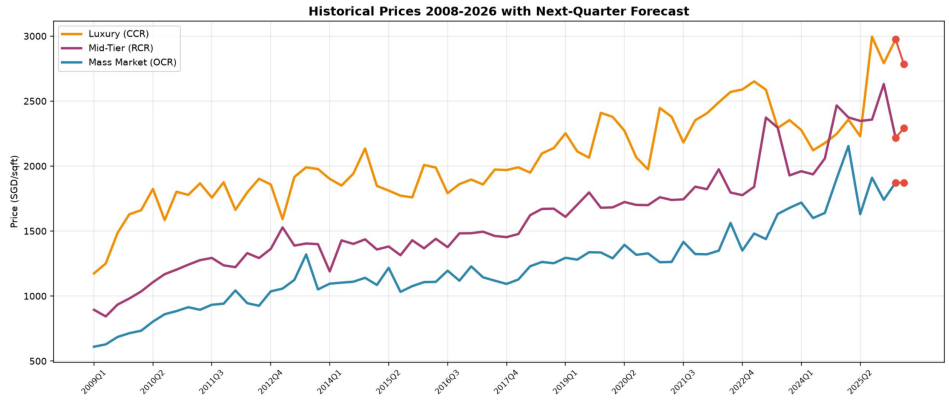
The system is designed to support analysis, not to operate as an autonomous valuation tool. Users can compare segment forecasts, review historical and forecast charts, inspect key explanatory variables, and ask the advisor for additional context. The dashboard and advisor provide these functions, while the uncertainty ranges come from rolling backtest residuals rather than Bayesian fusion.

House Price Forecast

Leak-Free | 28 Windows | L2 Direct

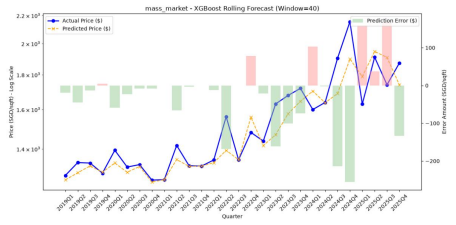
Mass Market Forecast	Mid-Tier Forecast	Luxury Forecast
$1,873 → $1,873	$2,217 → $2,291	$2,975 → $2,976
-0.02% | DirAcc 85.7%	+3.27% | DirAcc 75.0%	+0.02% | DirAcc 82.1%

Historical Prices & Next-Quarter Forecast

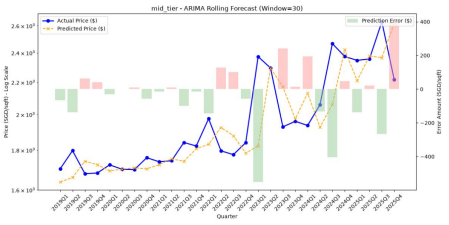


Red endpoints are next-quarter forecasts. Mass Market = flat, Mid-Tier rising, Luxury declining.

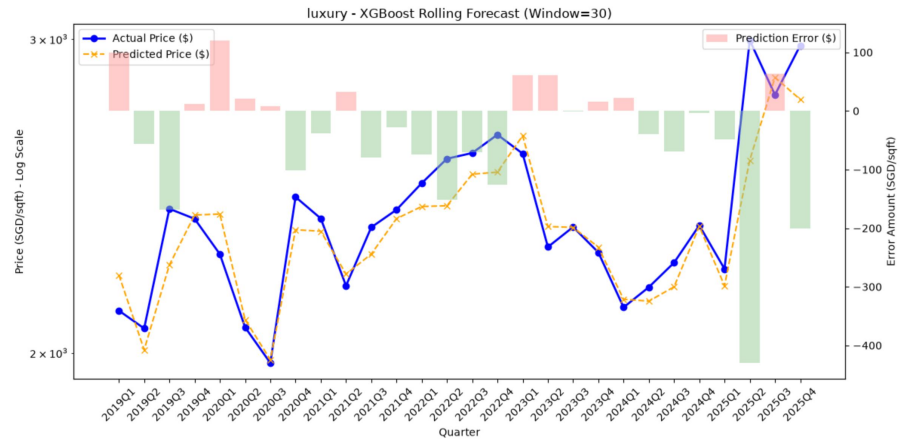
Mass Market: Actual vs Model Predictions



Mid-Tier: Actual vs Model Predictions



Luxury: Actual vs Model Predictions



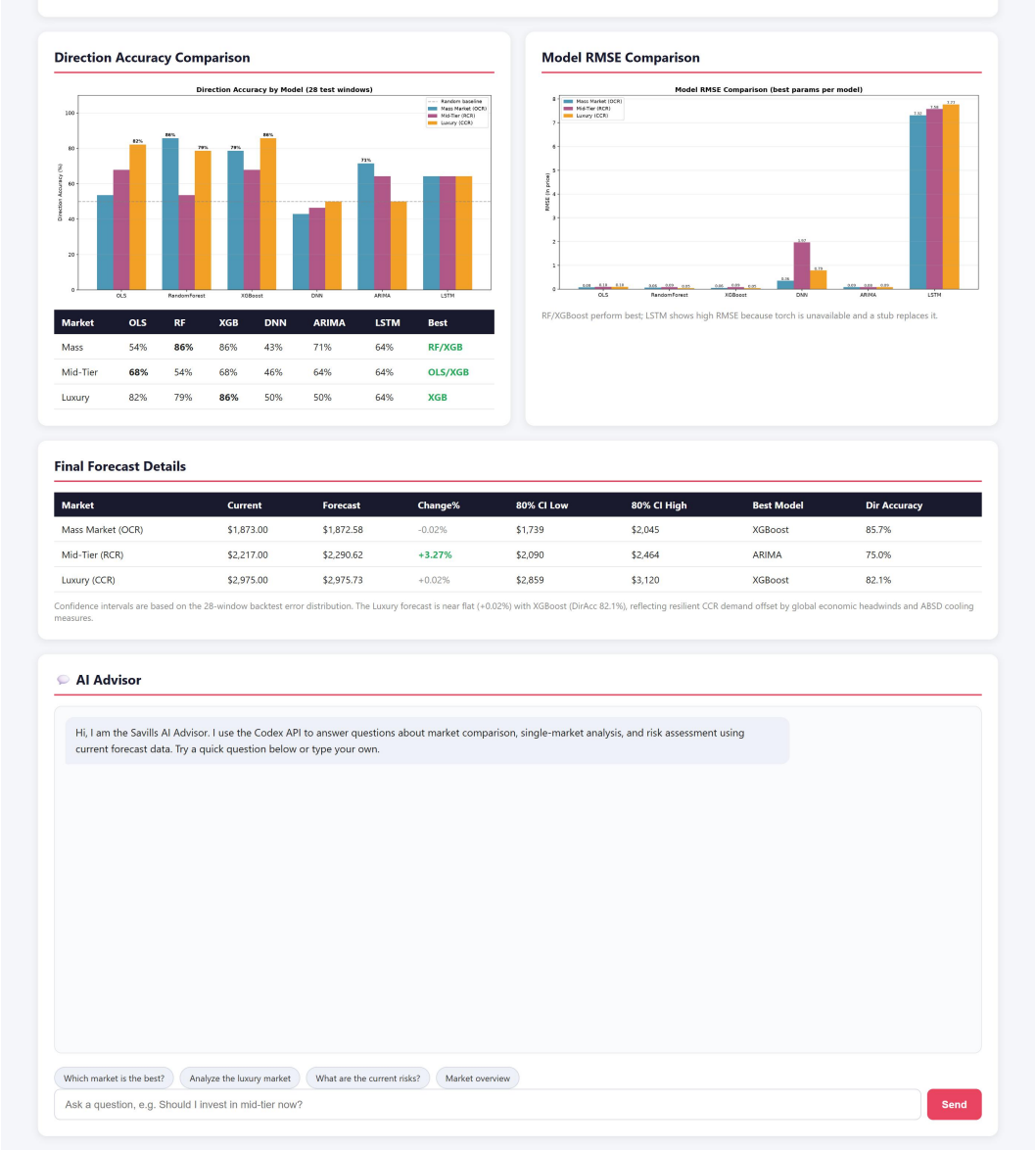


Figure 5 — Savills Property Forecasting System web dashboard. The delivered dashboard is taller than one printed page, so it is shown as two stacked panels: charts above, followed by the forecast table and AI Advisor.

8. Recommendations and Future Work

- Enforce the configured 2008Q1–2022Q4 training, 2023Q1–2024Q4 validation, and 2025Q1–2026Q1 blind-test split, and freeze all hyperparameters before the final evaluation.
- Populate and validate private_supply, or remove it from the active feature matrix rather than filling an entirely missing series with 0.0.
- Recompute the GDP × sentiment interaction after sentiment injection so that it reflects actual sentiment rather than GDP alone.

- Validate SORA and SGD NEER against authoritative source documentation before production use.
- Move the report-level ARIMA diagnostics (ADF and ACF/PACF in §4.3) into the pipeline so they are generated reproducibly, while retaining the explicit ARIMA(1,1,0)/ARIMA(2,1,0) benchmark over 30- and 40-quarter windows.
- Document and calibrate the residual-based interval method using genuinely out-of-sample residuals.
- Freeze a reproducible dataset, code version, model configuration, and output folder before generating final tables and figures.
- Improve LSTM by using genuine multi-quarter sequences and test whether this adds value beyond lag-based models; monthly or transaction-level data could also increase the effective sample size.
- Run an ablation study to test whether AI sentiment adds out-of-sample value, given its last-place SHAP ranking.

9. Conclusion

The Savills Property Forecasting System combines quarterly data preparation, AI-assisted sentiment, six forecasting models, time-aware evaluation, and explainability in one decision-support workflow. Its main contribution is not a single 'best' algorithm, but a segment-specific comparison that allows different models to win in different markets. In the final run, Random Forest performs best for the mass market, while XGBoost performs best for the mid-tier and luxury segments. AI sentiment has the lowest importance among the 15 retained features, and the deep models remain unstable on the small sample.

For Savills, the results indicate modest short-term upward momentum in the mass market, a smaller positive signal in the mid-tier segment, and a flat luxury outlook. These findings should be read together with the empirical prediction intervals and the limitations discussed in the report.

10. References

References from the team's literature review (ten studies).

1. Huang, W., & Zhang, Y. (2012). Structural Change Modeling of Singapore Private Housing Price in Simultaneous Equation Model. Journal of Financial Risk Management, 1(2), 7–14. https://doi.org/10.4236/jfrm.2012.12002

2. Xu, J. (2017). Hedonic Modeling of Singapore's Resale Public Housing Market. Honors thesis, Duke University. https://hdl.handle.net/10161/14268

3. Lim, W.-T., Wang, L., Wang, Y., & Chang, Q. (2016). Housing Price Prediction Using Neural Networks. In Proceedings of the 12th International Conference on Natural Computation, Fuzzy

Systems and Knowledge Discovery (ICNC FSKD 2016), pp. 518–522. IEEE. https://doi.org/10.1109/FSKD.2016.7603227

4. Wang, L., Chan, F. F., Wang, Y., & Chang, Q. (2016). Predicting Public Housing Prices Using Delayed Neural Networks. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), pp. 3589–3592. IEEE. https://doi.org/10.1109/TENCON.2016.7848726

5. Cao, K., Diao, M., & Wu, B. (2019). A Big Data–Based Geographically Weighted Regression Model for Public Housing Prices: A Case Study in Singapore. Annals of the American Association of Geographers, 109(1), 173–186. https://doi.org/10.1080/24694452.2018.1470925

6. Wang, Y., Cai, F., Cheng, S.-F., Wu, B., & Cao, K. (2022). Taxi Travel Time Based Geographically Weighted Regression Model (GWR) for Modeling Public Housing Prices in Singapore. In Proceedings of the 29th International Conference on Geoinformatics, pp. 1–5. IEEE. https://doi.org/10.1109/Geoinformatics57846.2022.9963833

7. Sing, T.-F., Yang, J. J., & Yu, S.-M. (2022). Boosted Tree Ensembles for AI Based Automated Valuation Models. Journal of Real Estate Finance and Economics, 65(4), 649–674. https://doi.org/10.1007/s11146-021-09861-1

8. Chia, W.-M., Li, M., & Tang, Y. (2017). Public and Private Housing Markets Dynamics in Singapore: The Role of Fundamentals. Journal of Housing Economics, 36, 44–61. https://doi.org/10.1016/j.jhe.2017.03.001

9. Durai, S. A. B., & Wang, Z. (2023). Resale HDB Price Prediction Considering Covid-19 Through Sentiment Analysis. In Proceedings of the 10th European Conference on Social Media (ECSM 2023). https://doi.org/10.34190/ecsm.10.1.1020

10. Liu, & Pan. (2025). Research of the Influence Factors of Housing Price — Take Singapore as an Example.

General methodological foundations (cited for method theory above; not among the ten team reviewed studies).

Rosen, S. (1974). Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition. Journal of Political Economy, 82(1), 34–55. https://doi.org/10.1086/260169

Box, G. E. P., & Jenkins, G. M. (1970). Time Series Analysis: Forecasting and Control. San Francisco: Holden Day. (Book; ISBN 978-0-8162-1094-7 — no DOI.)

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Shapley, L. S. (1953). A Value for n-Person Games. In H. W. Kuhn & A. W. Tucker (Eds.), Contributions to the Theory of Games II (pp. 307–317). Princeton University Press. https://doi.org/10.1515/9781400881970-018

Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. In Advances in Neural Information Processing Systems 30 (pp. 4765–4774). https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. (1999). Bayesian Model Averaging: A Tutorial. Statistical Science, 14(4), 382–401. https://doi.org/10.1214/ss/1009212519

Goodman, A. C., & Thibodeau, T. G. (2003). Housing Market Segmentation and Hedonic Prediction Accuracy. Journal of Housing Economics, 12(3), 181–201. https://doi.org/10.1016/S1051-1377(03)00031-7

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., & Zhou, D. (2022). Chain of Thought Prompting Elicits Reasoning in Large Language Models. In Advances in Neural Information Processing Systems 35 (NeurIPS 2022). https://arxiv.org/abs/2201.11903

Hyndman, R. J., & Athanasopoulos, G. (2021). Forecasting: Principles and Practice (3rd ed.). OTexts. https://OTexts.com/fpp3/

Bergmeir, C., & Benítez, J. M. (2012). On the Use of Cross Validation for Time Series Predictor Evaluation. Information Sciences, 191, 192–213. https://doi.org/10.1016/j.ins.2011.12.028

Peng, R. D. (2011). Reproducible Research in Computational Science. Science, 334(6060), 1226–1227. https://doi.org/10.1126/science.1213847

Full bibliographic details and a DOI or link are provided where a source could be verified. The venue and link for Liu & Pan (2025) should be confirmed before final submission.

Appendix A. Model Configurations and Hyperparameter Search Space

The configurations below reflect the final ml_trainer.py implementation used for the reported run.

Table 7. Model configurations and hyperparameter search space.

Model	Hyperparameters / search space	Source

Model	Hyperparameters / search space	Source
OLS	— (no hyperparameters)	ml_trainer.py
ARIMA	orders (1,1,0) and (2,1,0); windows 30, 40	ml_trainer.py
Random Forest	n_estimators ∈ {100, 200}; max_depth ∈ {5, 10}; windows 30, 40	ml_trainer.py
XGBoost	learning_rate ∈ {0.05, 0.1}; max_depth ∈ {3, 5}; windows 30, 40	ml_trainer.py
DNN	hidden_layer_sizes ∈ {(10,), (20,10)}; solver=lbfgs; windows 30, 40	ml_trainer.py
LSTM	hidden_size ∈ {8, 16}; epochs ∈ {150, 200}; windows 30, 40; sequence length 1	ml_trainer.py
Common	one-quarter-ahead horizon; RMSE primary, MAE + directional accuracy supporting; top-15 RF feature selection	config.py

Note: the RF/XGB/DNN parameter grids defined in config.py (RF_PARAM_GRID, XGB_PARAM_GRID, DNN_PARAM_GRID) are broader than the reduced search space actually executed by ml_trainer.py. Table 7 reports the configurations that were actually run.

Appendix B. Data Dictionary

Table 8. Data dictionary.

Feature	Description	Unit / transformation	Status

Feature	Description	Unit / transformation	Status
segment property price	Savills segment-level residential price index	log-transformed	verified
nominal GDP	Singapore quarterly nominal GDP	log-transformed, lags 1–4	verified
CPI	consumer price index (monthly, aggregated quarterly)	level	verified
unemployment	unemployment rate (annual, interpolated)	level	verified
SORA	Singapore Overnight Rate Average	level	verified
ABSD change	Additional Buyer's Stamp Duty policy change indicator	indicator	verified
STI index	Straits Times Index	level	verified
SGD NEER	Singapore dollar nominal effective exchange rate	level	verified
AI sentiment	L1 qualitative score (three analyst personas)	−5 to +5	verified
lag variables	price & GDP lags 1–4	engineered	verified
interactions	CPI × GDP; log GDP × AI sentiment	engineered	GDP × sentiment computed before injection; revise
private_supply	private housing supply	missing (0/73 rows)	populate or remove

Source: utils/config.py; data/external_features.csv; data/processed/feature_store.csv (69 rows × 35 columns)

Appendix C. Reproducibility and Versioning

Code version: Savills_AI_v4.0_project_code.

The delivered Savills_AI_v4.0_project code directory is not under git version control, so no commit hash or release tag is available. Earlier v3.0 material is documented in two markdown reports included in the package—Savills_AI_v3.0_Business_Presentation_Report.md and Forward_Looking_Report_20260727.md, but neither has a commit hash or version tag. Initialising git and tagging the final release is recommended before submission.

- Dataset snapshot: Book1(3).xlsx (Savills prices) + SingStat macro + external_features.csv.
- Data snapshot and source-file inventory (timestamps from the delivered package):
- Book1(3).xlsx — Savills segment-level price indices.
- data/macro_fallback.csv — nominal GDP, CPI, unemployment rate.
- data/external_features.csv — SORA, ABSD change, STI, private_supply, SGD NEER; 73 quarters, 2008Q1–2026Q1.
- data/policy_timeline.csv — quarterly policy events.
- data/processed/feature_store.csv — 69 rows × 35 features, 2009Q1–2026Q1.
- data/outputs/all_parameters_tested_*.csv and data/outputs/predictions.csv — final six-model results and forecasts.
- report_output/shap_importance.csv (and PNGs) — SHAP explainability output (2026-08-12).
- Top-level deliverables included with the code package:
- README.txt — package manifest listing all deliverables and the final three segment forecast.
- v4.0 _New_Plan_Remove_Bayesian.md — architecture note recording the four reasons for removing Bayesian fusion.
- Team_Research_Results.txt — literature summary supporting Section 4.7.
- dashboard.html, advisor_server.py, start_dashboard.bat — web dashboard and AI advisor delivery.
- chart1_mass_market.png, chart1_mid_tier.png, chart1_luxury.png, chart2_direction.png, chart3_rmse.png, chart4_history_forecast.png, predictions.csv — dashboard charts and final forecasts.

- Final outputs: data/outputs/all_parameters_tested_*.csv and predictions.csv; report_output/shap_importance.csv.
- Environment: Python and the delivered requirements.txt (pandas, statsmodels, scikit-learn, xgboost, torch, shap, google-genai, streamlit; pymc commented out).
- Random seeds in the final implementation (confirmed from source):
- RandomForest, XGBoost and DNN (MLPRegressor): random_state = 42 (models/ml_trainer.py; app.py).
- AI sentiment (L1): random.seed(42) and np.random.seed(42) (agents/sentiment_analyst.py).
- ARIMA: no seed — statsmodels ARIMA is estimated deterministically and takes no stochastic seed parameter.
- LSTM: no seed is set. The code does not call torch.manual_seed, so PyTorch weight initialisation is non-deterministic and LSTM results are not exactly reproducible across runs; this remains a reproducibility gap.