Generative AI use statement

I, Yue Zhang, hereby confirm that I used gamma.app to create ppt based on text. The tool(s) was/were used to provide presentation ppr. The output from the tool(s) was/were modified by me.

- 1. When using AI, I have ensured that the work produced is still my own and I understand that submitting unmodified output from a generative AI tool as my own is NOT acceptable. I understand that I am expected to build on the output, ensuring any submissions are my own ideas and knowledge.
- 2. I acknowledge awareness of any updates to the generative AI tools used, up to the date of this submission. This includes AI plug-ins or assistants included in existing programs, such as Grammarly. I take responsibility for any fabricated references or factual errors stemming from the use of these tools.
- 3. I have informed myself of the limitations and implications of using generative AI and related technologies, including the reinforcement of biases and propensity for fabrication.
- 4. I have used these tools ethically, including not uploading confidential, private, personal, copyrighted, or otherwise sensitive information.
- 5. To assist with maintaining academic integrity, I have appropriately acknowledged any use of generative AI in my work (list below as applicable).
- 6. I acknowledge that any undeclared use of generative AI will constitute academic dishonesty and will be dealt with according to relevant University policy.
- 7. I understand that I will be held accountable for any academic misconduct that arises in breach of any relevant University policy, as well as the consequences of such infringements.

Tool used:

https://gamma.app/

Date accessed:

Oct 13th, 2025

Prompt(s) entered:

NZ House Price Prediction System

Production ML System with End-to-End Pipeline

Team Members: Xuanhui, Ming, Zhengyang, Yue

Live Demo: app-test-qxq5b9dukmh7yw6xuyufxc.streamlit.app

Slide 1: Project Overview

What We Built

A complete machine learning system for forecasting New Zealand house price growth

Key Components

- Automated Data Pipeline ETL from multiple sources
- Feature Engineering 53 features from 8 base variables
- Multiple ML Models ETS, XGBoost, CatBoost
- Interactive Dashboard Streamlit web application
- Cloud Deployment Production-ready system

Target Variable

HPI Growth - Quarterly percentage change in House Price Index

Slide 2: Data Architecture

Data Sources

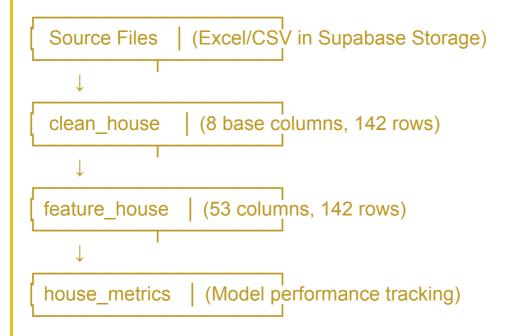
Source	Variables	Frequenc
		у

Housing Data	Sales, HPI, Stock, Investment	Quarterly
Economic Indicators	CPI, GDP	Quarterly
Monetary Policy	Official Cash Rate (OCR)	Quarterly

Infrastructure: SupabaseWhy Supabase instead of AWS?

- AWS-backed (uses S3 + RDS underneath)
- Simplified team collaboration
- Zero DevOps overhead
- Unified API for storage + database
- No account/billing complexity

Database Schema



Slide 3: Feature Engineering Strategy

The Challenge

Without Centralized Features:

- X Each member creates different features
- Naming conflicts & duplicates

- Risk of data leakage
- Deployment nightmare (different prep logic per model)

Our Solution: Common Feature Store

feature_house table = Single Source of Truth

53 Engineered Features

Temporal Features (Lags)

- 1-4 quarters, 4 years back
- Example: hpi_growth_lag1, house_sales_lag4

Rolling Aggregations

- 1-year, 4-year, 10-year windows
- Example: ocr_rolling_mean_1y, cpi_rolling_mean_4y

Derived Features

- Growth rates, differences, ratios
- Example: hpi_growth_diff_lag1_minus_lag4

Policy Indicators

- COVID lockdown period (2020 Q2-Q3)
- Reopening/supply constraints (2021 Q2 2022 Q4)

Slide 4: Model Development

Three Complementary Approaches

- 1. ETS (Exponential Smoothing) Xuanhui
 - Classical time-series forecasting
 - Trend + Seasonal components
 - Baseline univariate approach

2. Tree-Based Models - Xuanhui & Ming

- XGBoost: Gradient boosting with custom features
- CatBoost: Optimized for categorical data
- Random Forest: Ensemble decision trees

3. Linear Models - Zhengyang

- Ridge Regression: L2 regularization
- Feature selection for linear assumptions
- Interpretability focus

Unified Training Framework

python

trainer.py - Base class for all models class BaseTrainer:

- load feature house()
- prepare_supervised() / prepare_univariate()
- fit() / predict_split()
- compute_metrics()
- upsert metrics()

Benefits:

- Consistent train/test splitting
- Standardized evaluation metrics
- Automatic metric logging to database

Slide 5: Interactive Dashboard

Live Demo Features

[Open app during presentation]

1. Data Viewer

- Browse clean_house / feature_house tables
- Download CSV exports
- Column information & date ranges

2. Exploratory Data Analysis

- Feature importance (F-statistic)
- Target distribution & time-series
- Correlation analysis
- Missing value reports
- Summary statistics

3. Model Training Interface

- Select models: ETS XGBoost CatBoost
- Configure hyperparameters (JSON)
- Choose features dynamically
- Set train/test split ratio
- Missing data strategy (drop/impute)

4. Performance Comparison

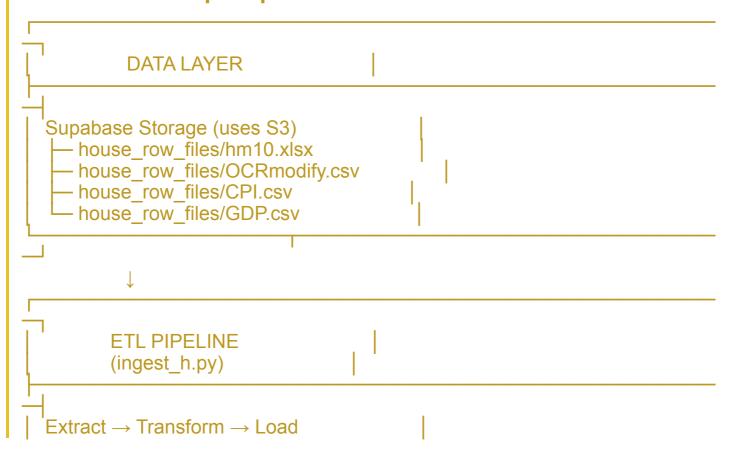
- Side-by-side metrics table
- Interactive forecast charts
- Train vs Test RMSE visualization

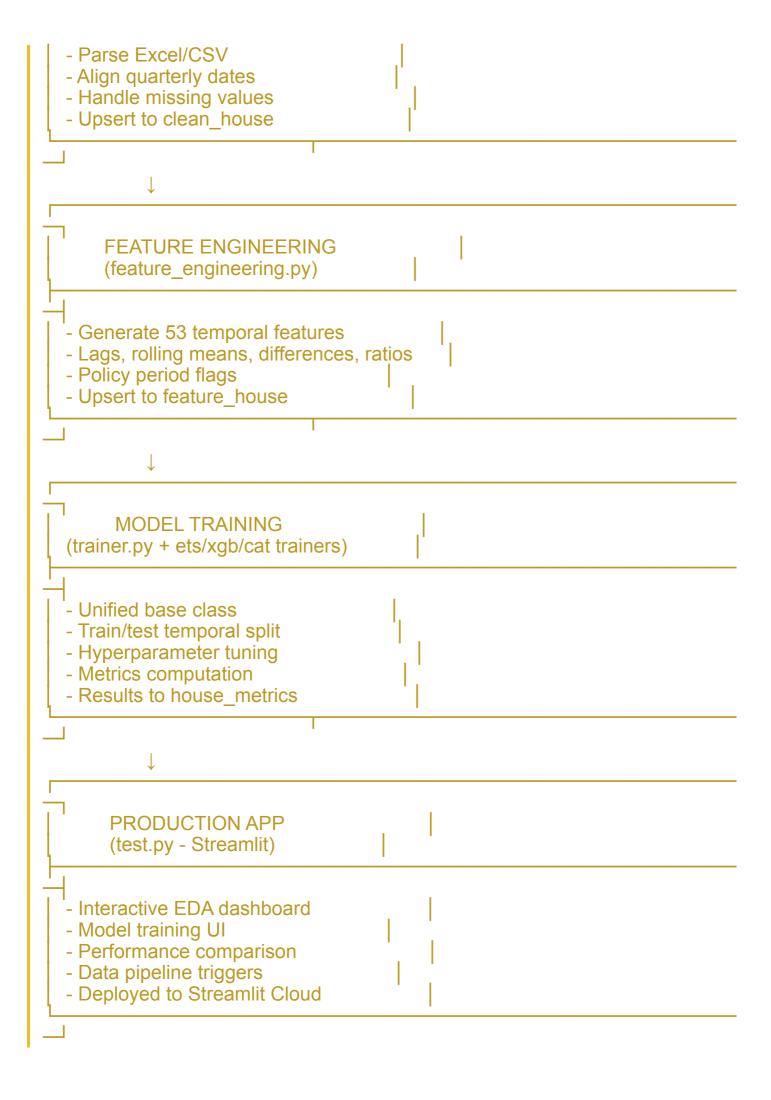
5. Data Pipeline Controls

- Trigger ETL refresh
- Run feature engineering
- Real-time status updates

Slide 6: System Architecture

End-to-End MLOps Pipeline





Slide 7: Key Technical Achievements

1. Production-Grade Infrastructure

- Automated ETL supporting continuous data updates
- Centralized feature store preventing inconsistencies
- Modular architecture enabling easy model addition

2. Collaborative Development

- Shared Supabase environment (no local DB setup)
- Consistent feature access for all team members
- Version-controlled codebase

3. Interactive Deployment

- Live web application (not just notebooks)
- Real-time model training and comparison
- Accessible to non-technical stakeholders

4. MLOps Best Practices

- Temporal train/test splitting (no data leakage)
- Automated metric tracking
- Reproducible experiments
- Missing data handling strategies

Slide 8: Model Performance

Evaluation Metrics

- RMSE (Root Mean Squared Error) lower is better
- MAE (Mean Absolute Error)
- R² (Coefficient of Determination)
- Train vs Test comparison

Sample Results

Model	Test RMSE	Test R ²	Features Used
ETS	~X.XX	~0.XX	Univariate (HPI growth only)

XGBoost	~X.XX	~0.XX	Top 8 selected features
CatBoost	~X.XX	~0.XX	Top 8 selected features
Ridge	~X.XX	~0.XX	Linear-selected features

[Live demo showing actual metrics from the app]

Feature Importance Insights

- Most important: hpi_growth_lag1, house_sales_lag1
- Economic indicators: OCR and CPI rolling means
- Policy flags show impact of COVID period

Slide 9: Challenges & Solutions

Challenge 1: AWS Complexity

Problem: Steep learning curve, account limitations, billing concerns **Solution:** Pivoted to Supabase (AWS-backed but developer-friendly) **Result:** Faster development, easier collaboration

Challenge 2: Feature Consistency

Problem: Risk of different feature implementations per team member **Solution:** Centralized feature_house table as single source of truth **Result:** Fair model comparisons, simplified deployment

Challenge 3: Temporal Data Alignment

Problem: Multiple data sources with different date formats **Solution:** Standardized quarter-end date parsing in ETL **Result:** Clean temporal joins, no misalignment

Challenge 4: Missing Data

Problem: CPI/GDP data not available for early periods **Solution:** Multiple strategies (drop, impute, forward-fill) + configurable UI **Result:** Robust handling, user choice in trade-offs

Slide 10: Novel Contributions

What Makes This Project Different?

1. Complete System, Not Just Models

- Most projects: Jupyter notebooks with model experiments
- Our project: Production system with ETL, feature store, deployment

2. Team Collaboration Infrastructure

- Feature store enabling independent model development
- Shared database eliminating coordination overhead

3. Interactive MLOps

- Live model training and comparison
- Data pipeline triggers from UI
- Real-time metric tracking

4. Pragmatic Engineering

- Chose right tools (Supabase) over complex ones (raw AWS)
- Balanced academic rigor with practical deployment

5. Industry Best Practices

- Feature store pattern (like Feast, Tecton)
- Unified trainer interface
- Temporal validation

Prompt output(s):

the ppt that I can export

Modification for assessment:

change some layout and typo