House Price Prediction in King County (2014–2015)

Team Project | Exploratory Data Analysis | Baselines | Feature Engineering | Model Comparison

Kaan Tirpanci & Anirudh Unni

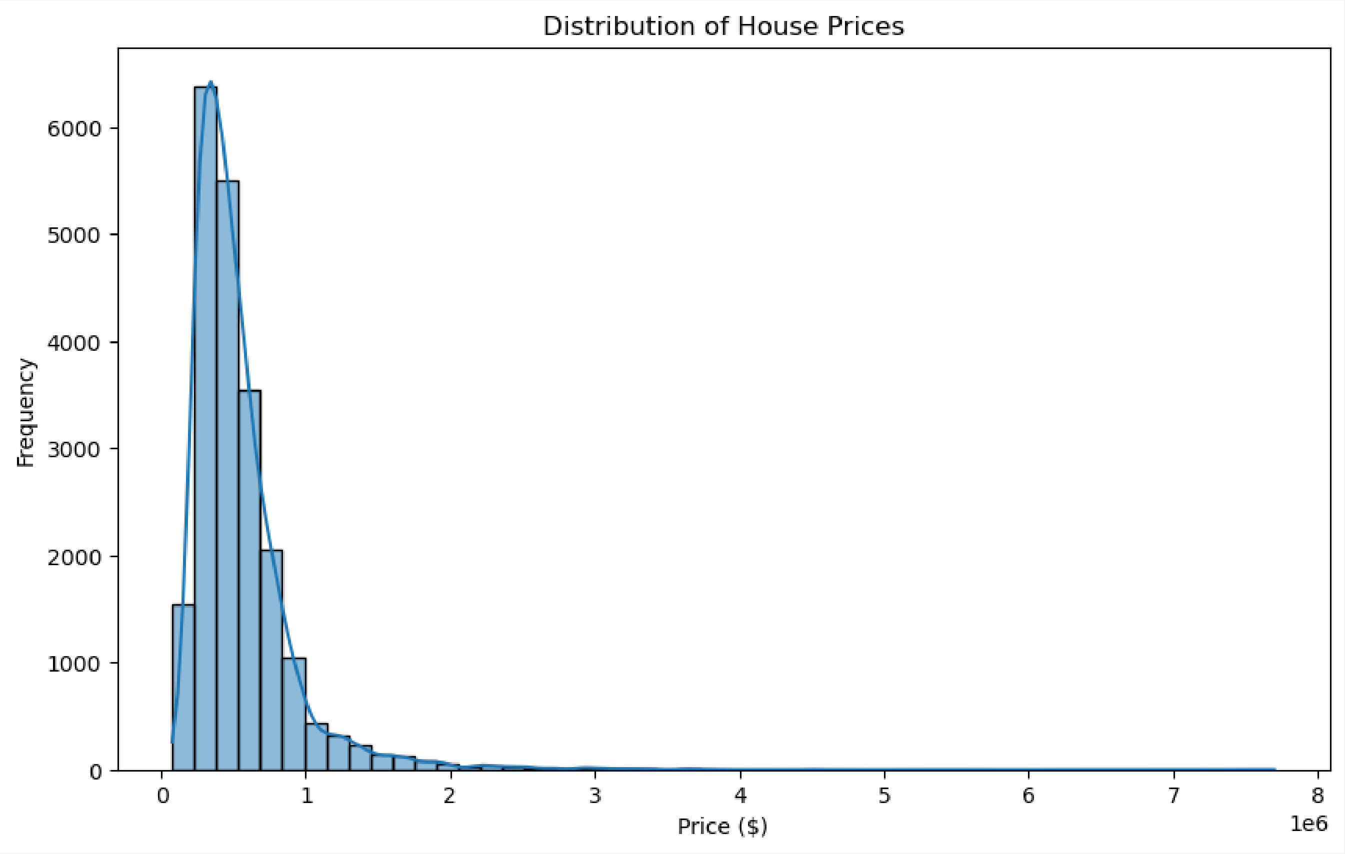
Dataset Overview

- 21 features, ~21K records
- Target: Price (House Sale Price)
- Timeframe: May 2014 – May 2015
- Geography: King County, including Seattle

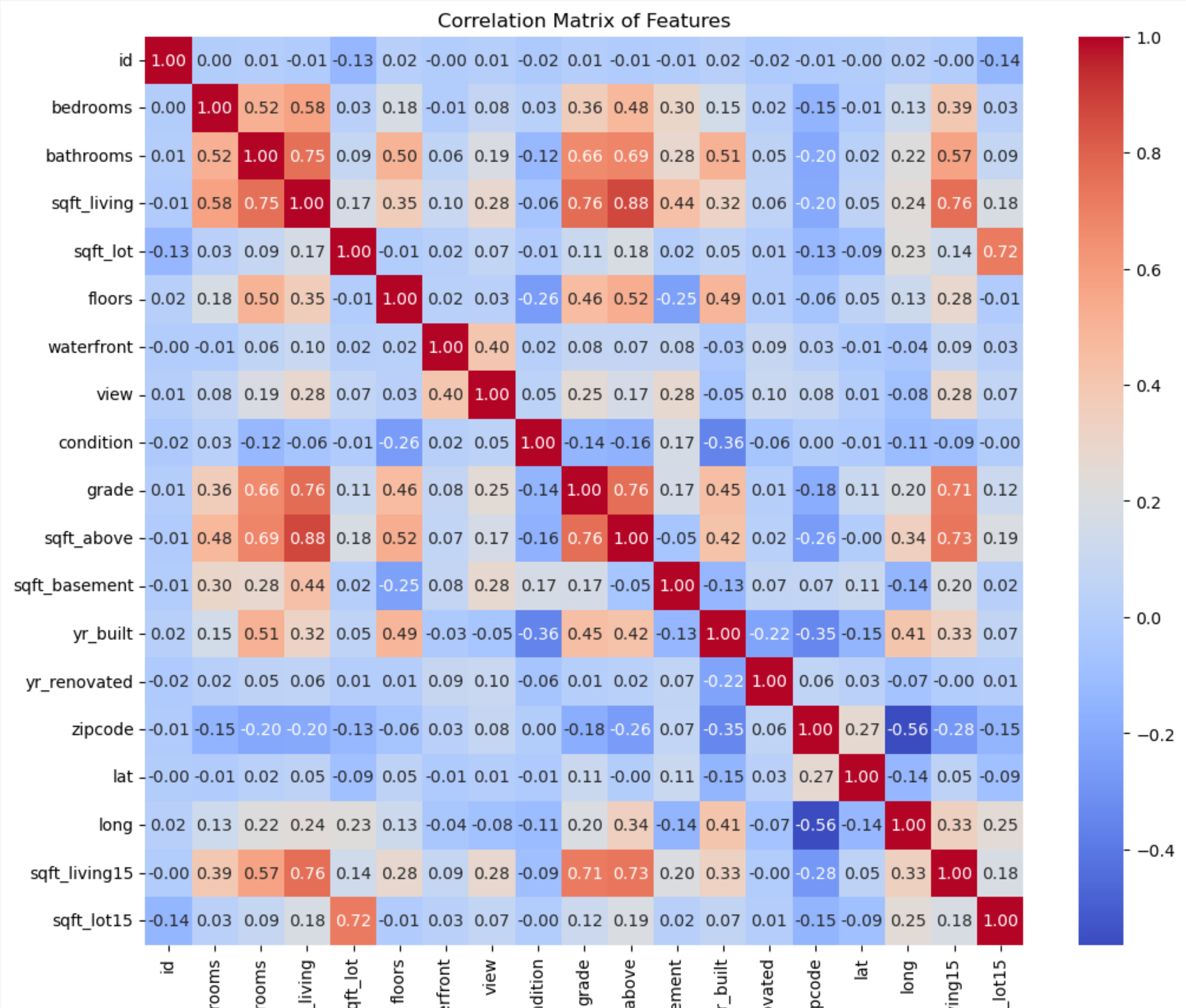
Business/Analytical Question

- Which features drive house prices?
- How accurately can we predict house prices using ML models?
- Learning goals: EDA, Baselines, Feature Engineering, Model Comparison, Hyperparameter Tuning

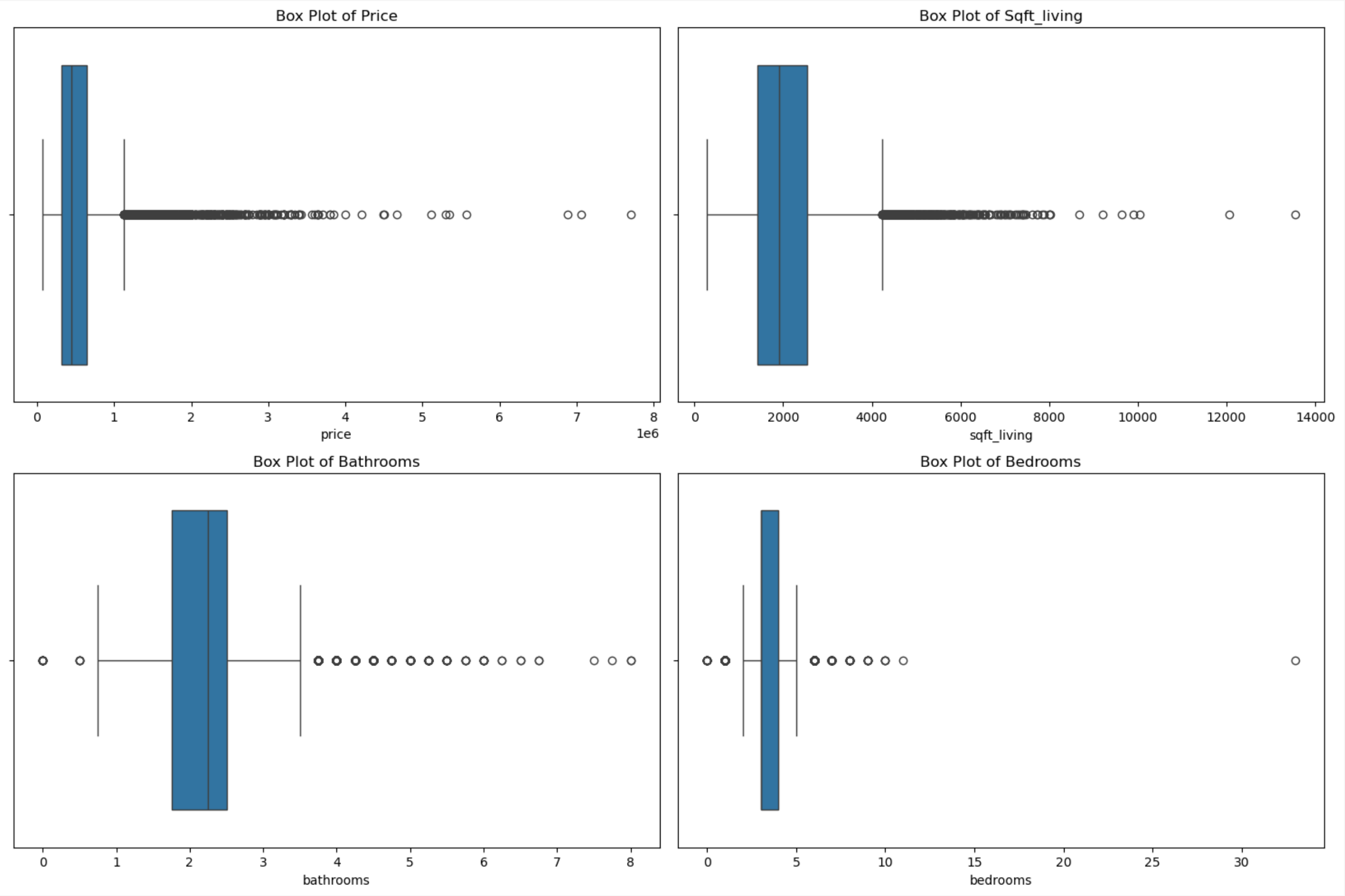
Exploratory Data Analysis: Price Distribution



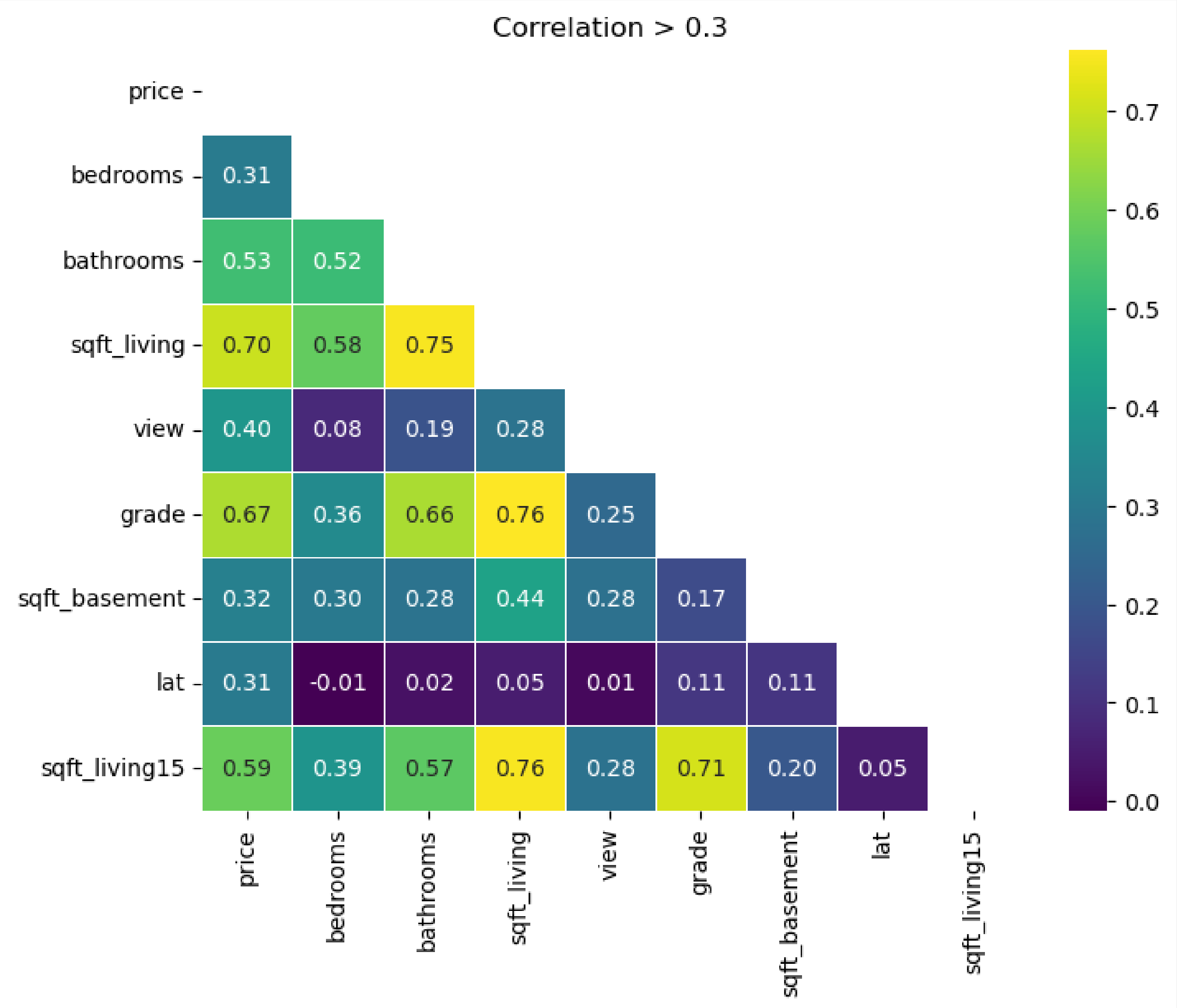
Exploratory Data Analysis: Feature Correlations



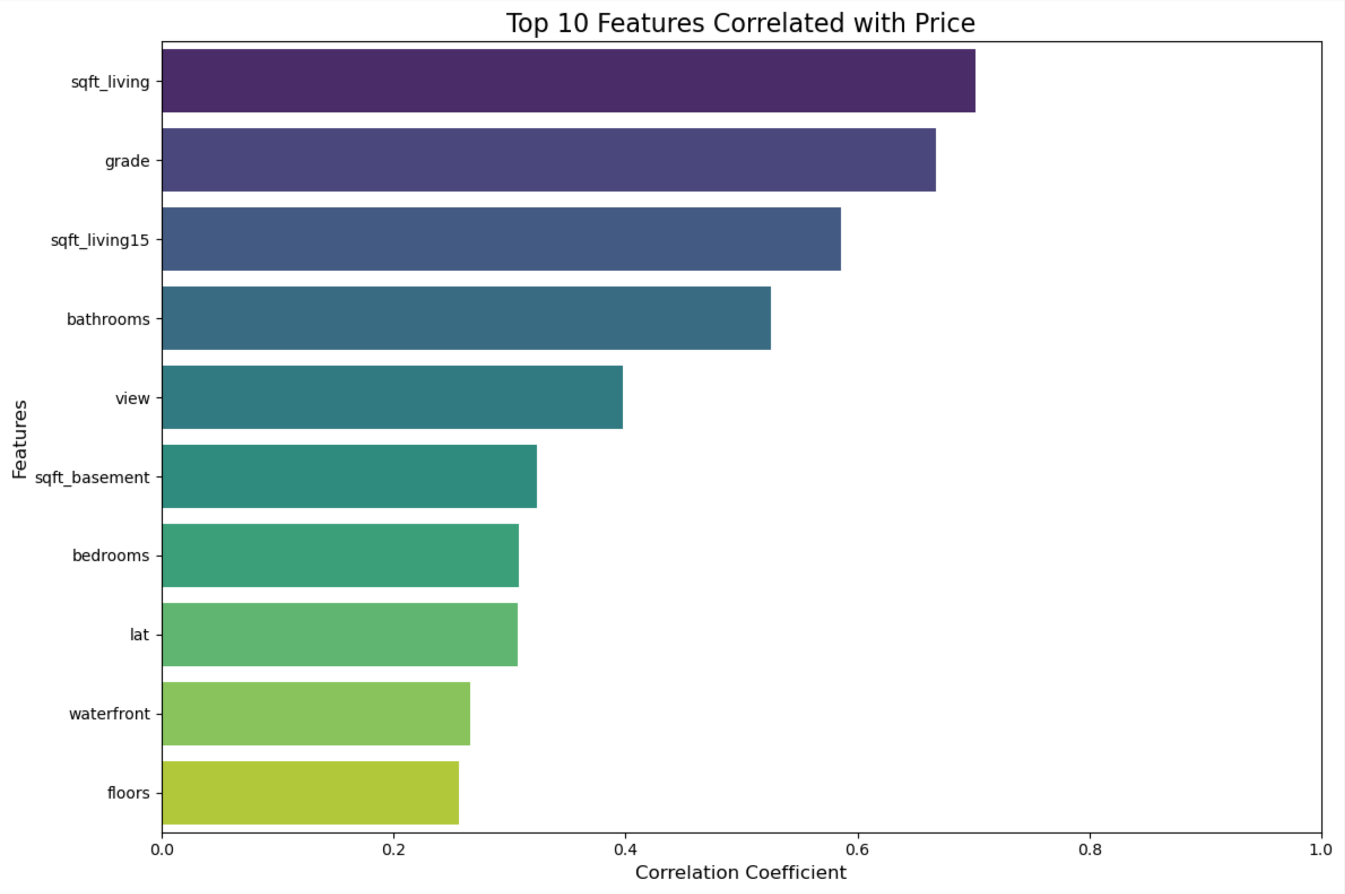
Outlier detection



Exploratory Data Analysis: Correlation Matrix



Bar Plots from Correlation Matrix



Model Performance (Baseline vs Feature Engineered)

Model	R^2 Train (Baseline)	R^2 Test (Baseline)	RMSE Test (Baseline)	R^2 Train (Engineered)	R^2 Test (Engineered)	RMSE Test (Engineered)
XGBoost	0.976	**0.893** 🟢	**112,572** 🟢	0.872	**0.865** 🟢	142,874
Random Forest	0.981	**0.890** 🟢	114,416	0.873	**0.854** 🟢	148,765
Gradient Boosting	0.901	0.866	126,018	0.873	**0.862** 🟢	144,755
Decision Tree	0.999 🔴	0.778 🔴	162,746 🔴	0.743	0.729 🔴	202,517 🔴
Ridge Regression	0.701	0.695	190,451	0.807	0.807 🟢	170,526
Lasso Regression	0.701	0.695	190,473	0.807	0.808 🟢	170,364
Linear Regression	0.701	0.695	190,473	0.807	0.808 🟢	170,368
KNN Regression	0.686 🔴	0.479 🔴	249,067 🔴	0.768	0.727	203,144
AdaBoost	0.389 🔴	0.284 🔴	291,758 🔴	0.200 🔴	0.156 🔴	357,224 🔴

🟢 = Best values (good generalization / lowest error)
🔴 = Overfitting or poor performance

Key Takeaway: Overfitting/Underfitting Insights

- **Baseline (no feature engineering):**
  - Ensemble models: Train $R^2 \approx 0.9$ → very high.
  - Linear models: Train $R^2 \approx 0.7$
  - Test R^2 much lower → **overfitting** is severe, especially for most Ensemble Methods Decision Tree, KNN, AdaBoost.

- **After feature engineering:**
  - **Linear/Ridge/Lasso:** Balanced Train vs Test R^2 (~0.807–0.808) → **better generalization**.
  - **Tree Ensembles (XGBoost, RF, GBM):** Train R^2 drops closer to Test R^2 → less overfitting, but slight test performance drop.
  - **Decision Tree:** Still weak (overfit + high RMSE).
  - **KNN:** Improves Test R^2 from 0.48 → 0.73; scaling helped.
  - **AdaBoost:** Worst performer, both underfitting & poor accuracy.

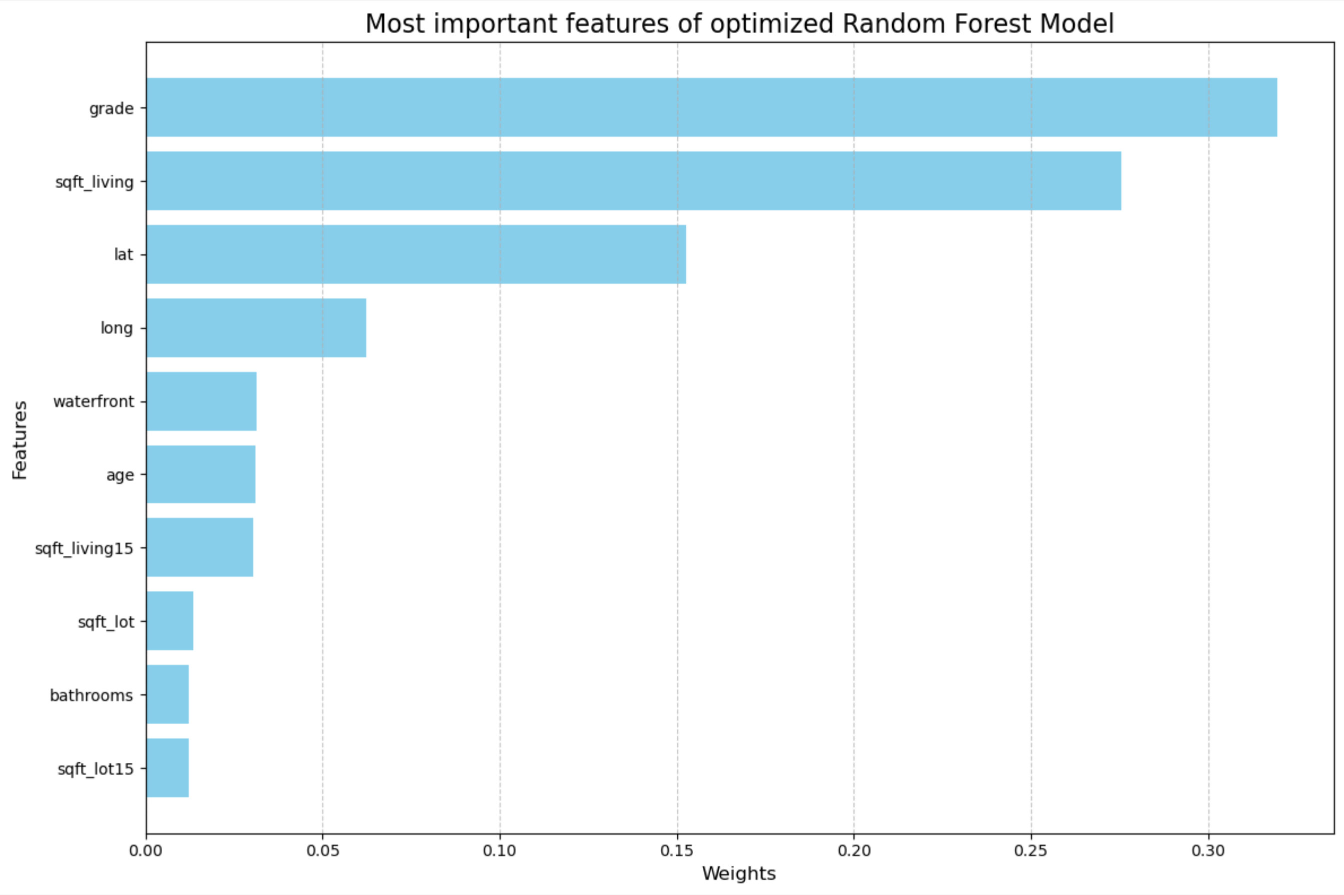
Feature engineering improves generalization for linear models, while powerful ensembles remain best overall predictors.

Hyperparameter Tuning: ADA Boost Model

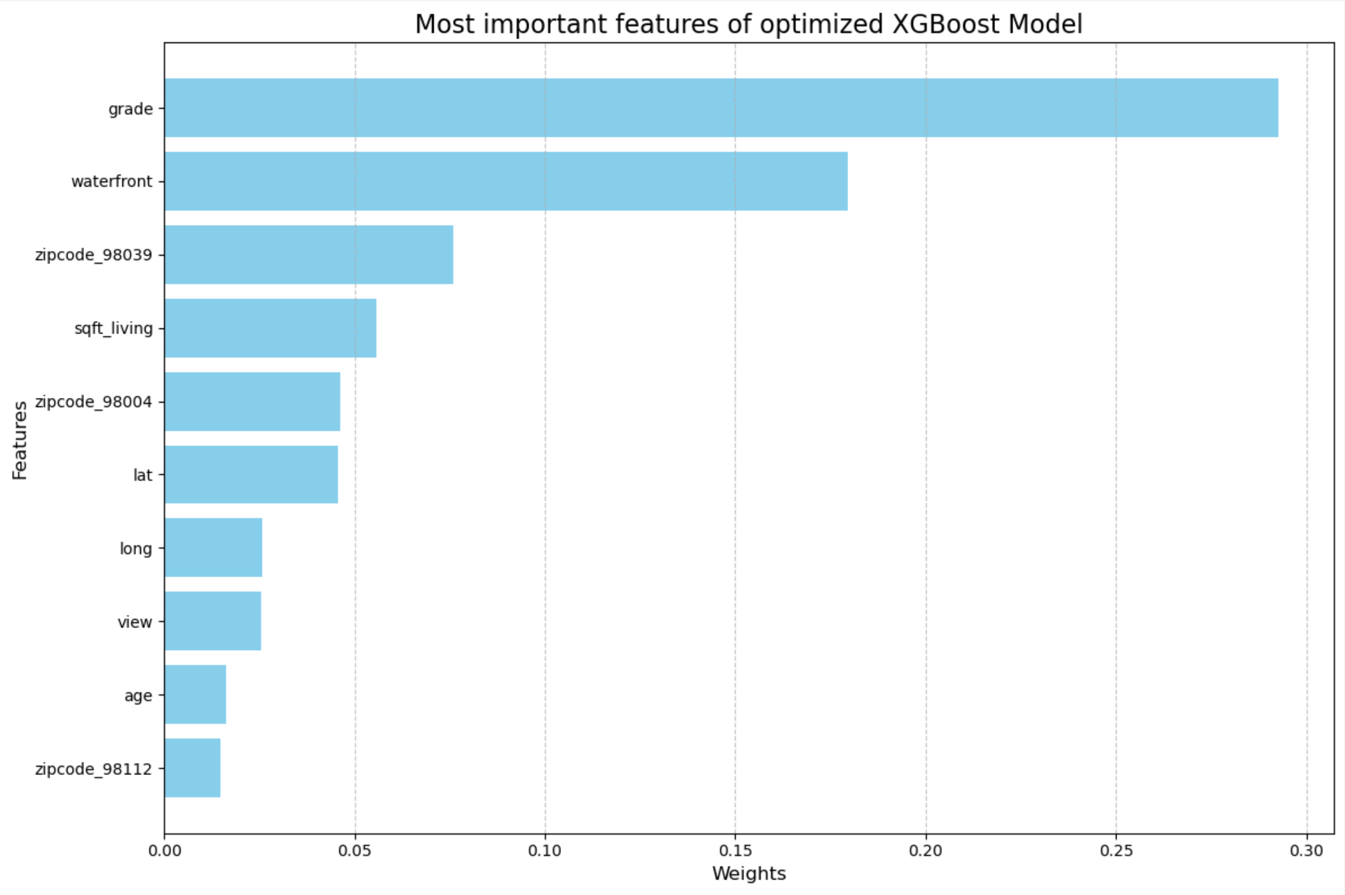
```
--- Tuning AdaBoost Regressor ---
Fitting 3 folds for each of 9 candidates, totalling 27 fits
Best AdaBoost Params: {'learning_rate': 0.1, 'n_estimators': 50}
Best AdaBoost CV RMSE: 202429.3564
```

```
--- Final Performance on Test Set ---
Final RMSE on Test Set: 227251.2322
Final R2 Score on Test Set: 0.6584
```

Hyperparameter Tuning: Random Forest Model



Hyperparameter Tuning: XGBoost Model



Key Takeaways

- Top drivers: grade, sqft_living, location (lat/long, zipcode)
- Tree-based models outperform linear models
- Feature engineering + hyperparameter tuning improves model reliability (less overfitting)

THANK YOU