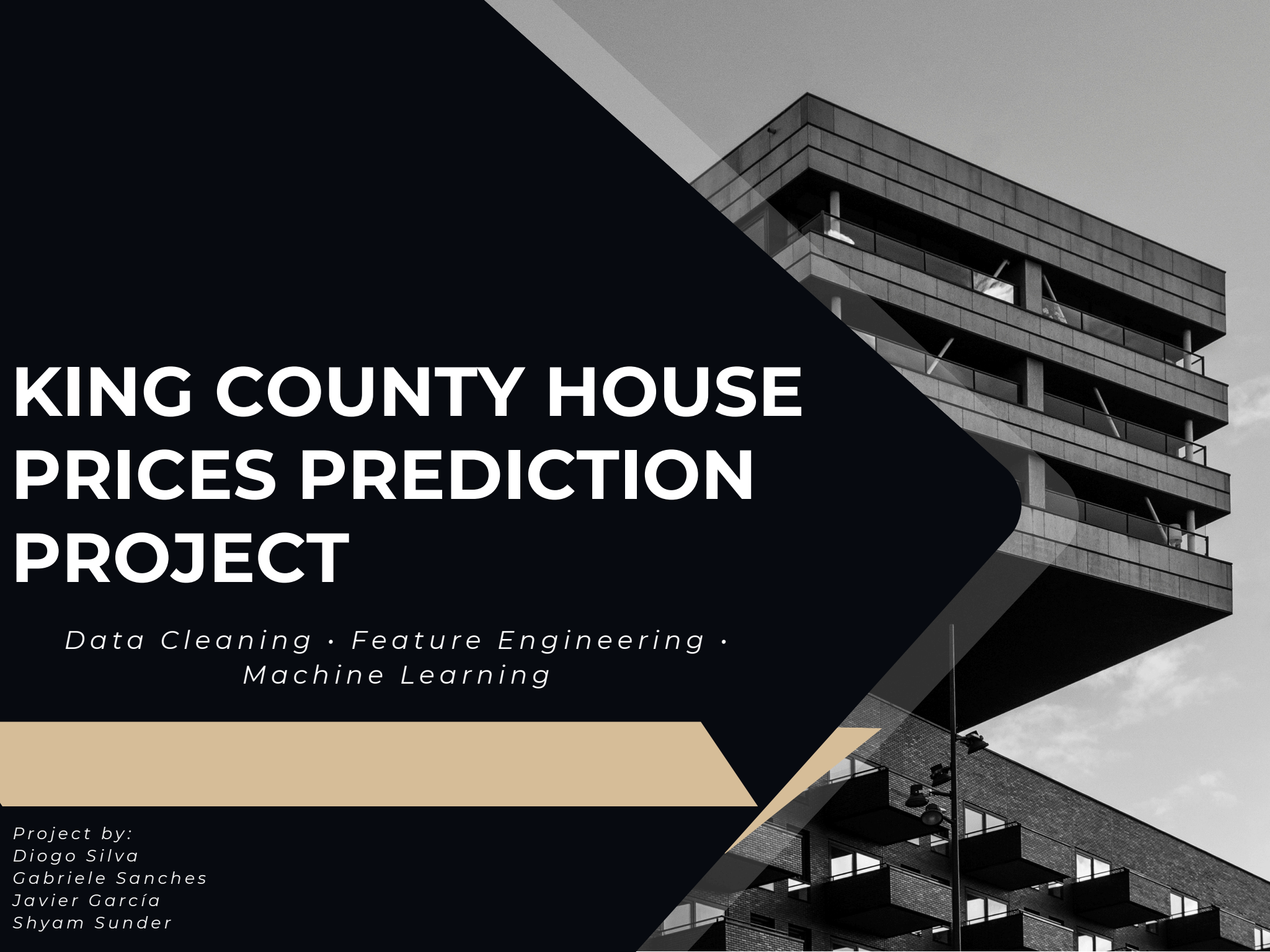
KING COUNTY HOUSE PRICES PREDICTION PROJECT

Data Cleaning · Feature Engineering · Machine Learning



Project by:
Diogo Silva
Gabriele Sanches
Javier García
Shyam Sunder

PROJECT GOAL AND DATA OVERVIEW

Project Goal

To build a regression model that accurately predicts house prices based on their features, such as size, location, and quality

The Data

We analyzed 21,613 sales from King County, WA (May 2014 - May 2015), examining 21 unique features for each property

Initial Challenge

The ‘price’ data was highly right-skewed with significant outliers. This required careful preprocessing and outlier management to build a reliable model.

DATA CLEANING STEPS

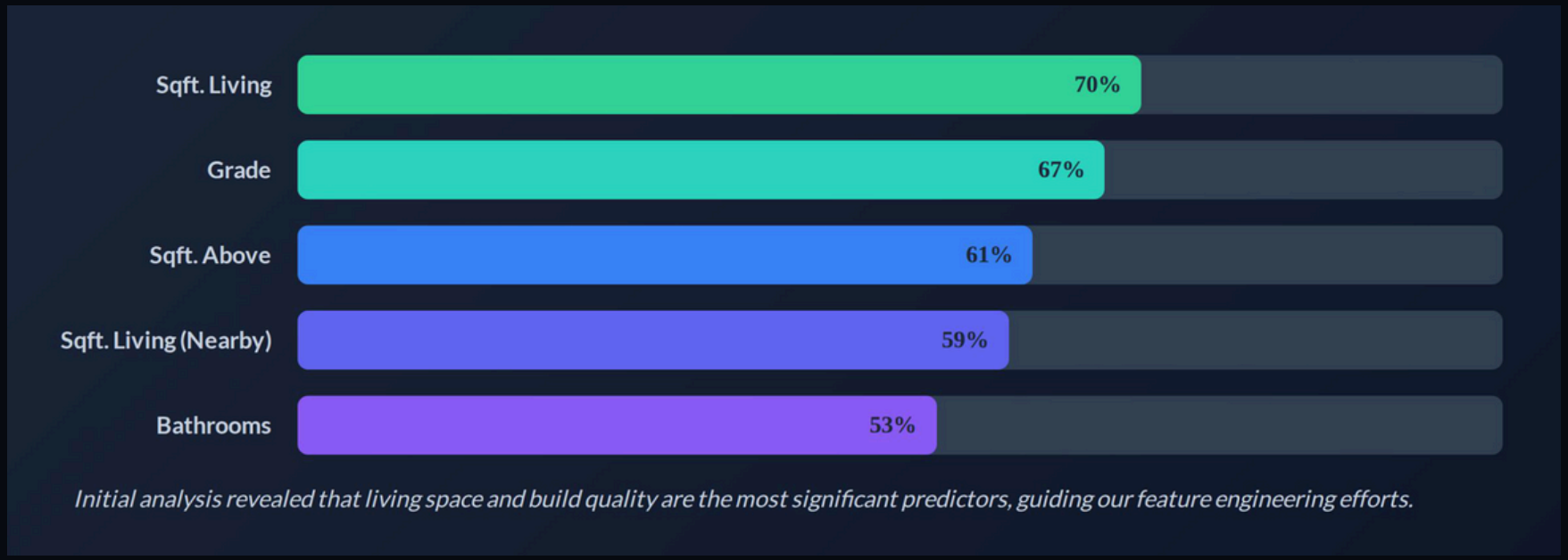
No Missing Values: The workflow began with a check for missing data, duplicates, etc. After the exploration was concluded, the dataset was complete with no further imputes needed.

Invalid Data: 16 rows containing 0-value entries for ‘bedroom’ & ‘bathrooms’ as they represent invalid data points.

Outlier Treatment: To stabilize the model, we filtered out the top and bottom 1% of prices using the IQR method to identify extreme outliers and removed them.

Feature Selection: Non-predictive features like ‘id’ and ‘date’ were dropped to focus the model on relevant information.

KEY PRICE DRIVERS (INITIAL EDA



Initial analysis revealed that living space and build quality are the most significant predictors, guiding our feature engineering efforts.

BASELINE MODELS

TRAINING BASELINE MODELS & IMPROVEMENT

Model Selection

We benchmarked a range of models, moving from simple baselines to more complex and powerful ensemble methods that can capture non-linear relatioships

- Linear Regression & KNN
- Random Forest
- XGBoost
- Model Improvement

Feature Engineering

We created new features to add context and capture relationships that the raw data missed:

- **house_age:** Calculated from yr_built
- **was_renovated**: A binary flag for any renovation
- **bath_per_bed:** A ratio to measure functionality
- **living_lot_ratio:** Ratio of living space to lot size

PERFORMANCE IMPROVEMENT

MODEL-SIDE

Scaling & Encoding

Data was prepared to be suitable for all model types:

- **RobustScaler**: Applied to normalized features, which minimizes the effect of remaining outliers
- **One-Hot Encoding**: We tested ‘zipcode’ as a categorical feature, which improved model accuracy over treating it as a number.

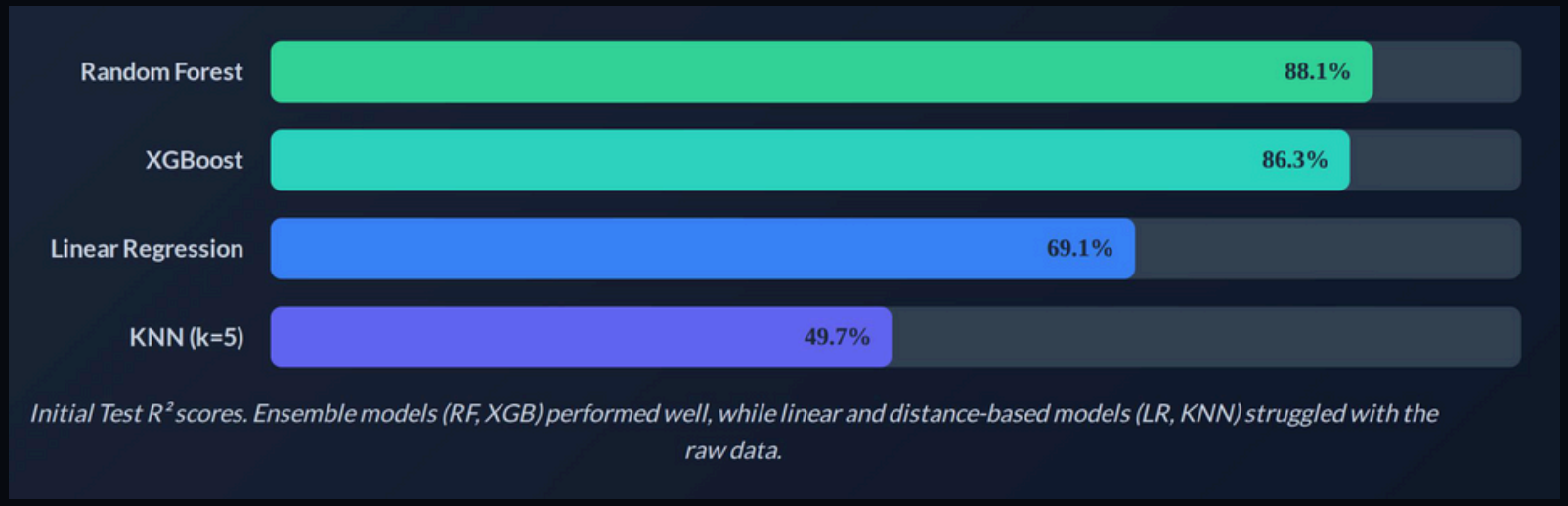
Hyperparameter Tuning

Used GridSearchCV to optimize Random Forest & XGBoost settings.

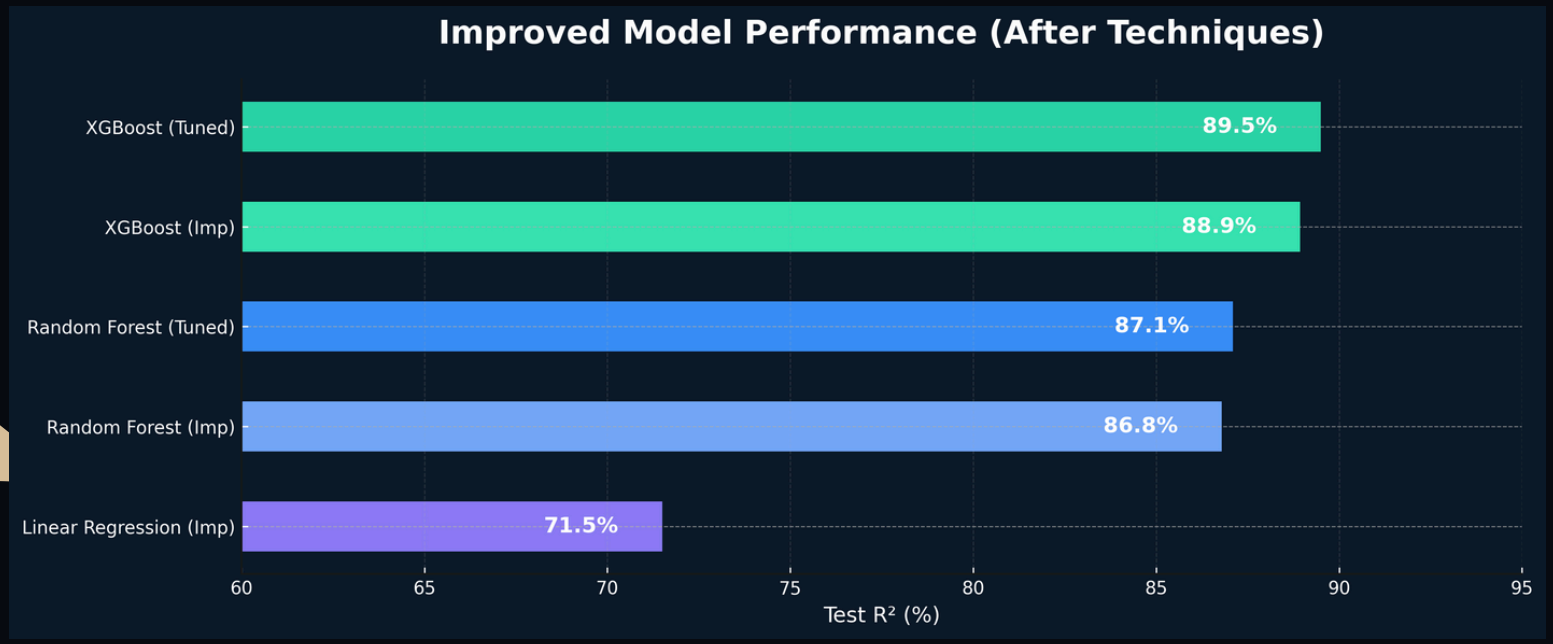
Tuned parameters: n_estimators, max_depth, and learning_rate to boost predictive accuracy.

MODEL PERFORMANCE EVOLUTION

BASELINE PERFORMANCE (BEFORE IMPROVEMENTS)



Initial Test R^2 scores. Ensemble models (RF, XGB) performed well, while linear and distance-based models (LR, KNN) struggled with the raw data.



FINAL MODEL COMPARISON

Model	Test R^2 (Higher is better)	Test RMSE (Lower is better)
XGBoost (Tuned)	0.895	$93,293
XGBoost (Improved)	0.889	$95,646
Random Forest (Tuned)	0.871	$103,346
Random Forest (Improved)	0.868	$104,570
Linear Regression (Improved)	0.715	$153,633
Linear Regression (Baseline)	0.691	$200,396
KNN (Baseline)	0.497	$255,854

Tuned XGBoost provided the best balance of high accuracy (89.5% R^2) and lowest prediction error ($93k RMSE).

BEST MODEL & TOP PRICE DRIVERS

Best Model & Top Price Drivers

Excelled at handling non-linear relationships and complex feature interactions.

89.5%

R^2 Score

$93K

RMSE

Top 5 Price Drivers

1. **Grade**
   Overall building quality
2. **Sqft. Living**
   Size of the home
3. **Latitude**
   Location factor
4. **Longitude**
   Location factor
5. **Sqft. Living 15**
   Neighborhood quality

INSIGHTS & CONCLUSION

Key Takeaways

- Feature Engineering was critical. ‘House Age’ and ‘Was Renovated’ provided significant value.
- Ensemble models (XGBoost, Random Forest) were highly effective for this dataset’s complexity.
- Location (‘lat’, ‘long’) and build quality (‘grade’) are as important, or even more so, than just size.

Key Challenges

- Managing extreme outliers in ‘Price’ and ‘Bedrooms’ was essential to prevent model bias.
- High multicollinearity (e.g., ‘sqft_living’ vs. ‘sqft_above’) required robust models like Ridge or tree-based ensembles to handle effectively.


THANK YOU