End-to-End Machine Learning Pipeline for Real Estate Valuation

Nawaraj Paudel, PhD
(Quantitative Modeling of Materials)
Data Scientist and ML Engineer

Real estate tech: Where massive markets meet innovation

Real Estate Market by the Numbers

- 146 million residential units valued at 43 trillion USD
- Commercial real estate valued at 21 trillion USD
- 2- 8 % of these properties sold every year

Marketcap Comparison

- S &P 500 : 45 Trillion
- NASDAQ 100 : 20 Trillion

Tech Transformation: Key Real Estate Domains

- PropTech: Real estate markets
- ConTech: Construction startups
- SmartRealEstate: Intelligent cities and buildings
- RealEstateFinTech: Mortage marketplace, Blockchain and smart contracts, Crowdfunding platforms
- Collaborative economy

The housing data contains 80 features including 43 categorical features

Dwelling Characteristics

- **OverallQual:** Rates the overall material and finish of the house
- **YearBuilt:** Original construction date
- **Year Remod/Add:** Remodel date

Living Area

- **1st Flr SF**: First Floor square feet
- **2nd Flr SF**: Second floor square feet
- **Low Qual Fin SF**: Low quality finished square feet (all floors)
- **Gr Liv Area**: Above grade (ground) living area square feet

Bedrooms and Bathrooms

- **Bsmt Full Bath:** Basement full bathrooms
- **Bsmt Half Bath:** Basement half bathrooms
- **Full Bath:** Full bathrooms above grade
- **Half Bath:** Half baths above grade
- **Bedroom:** Bedrooms above grade

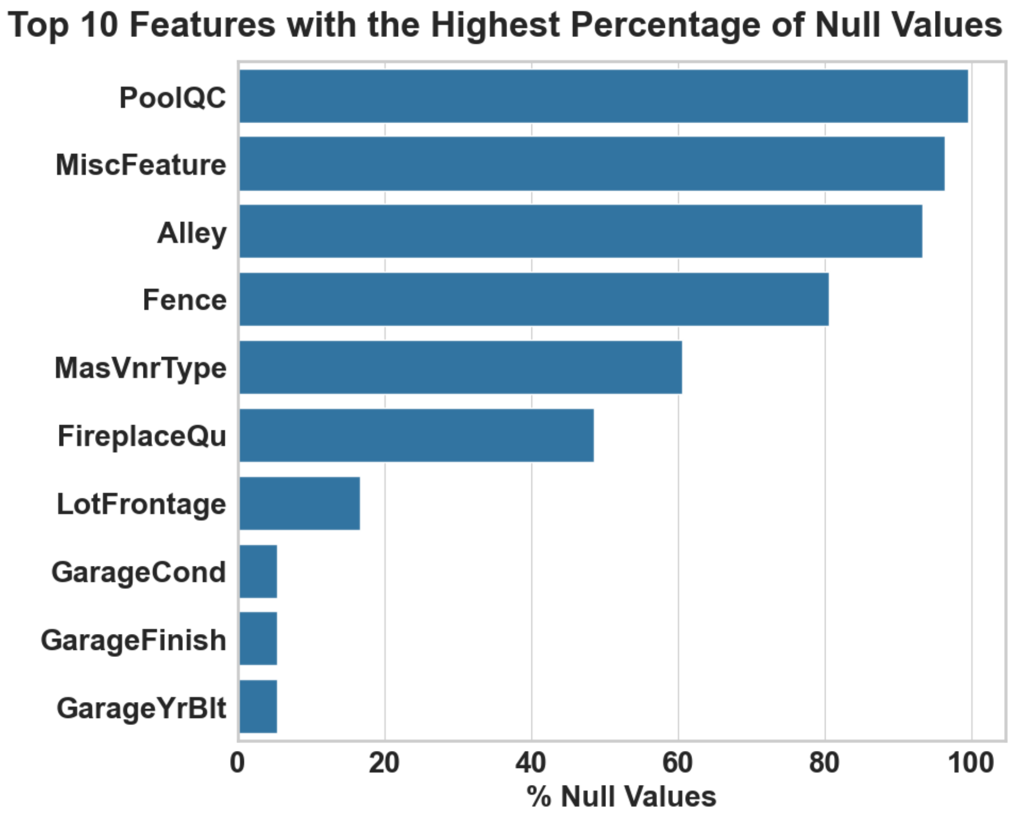
Other Features

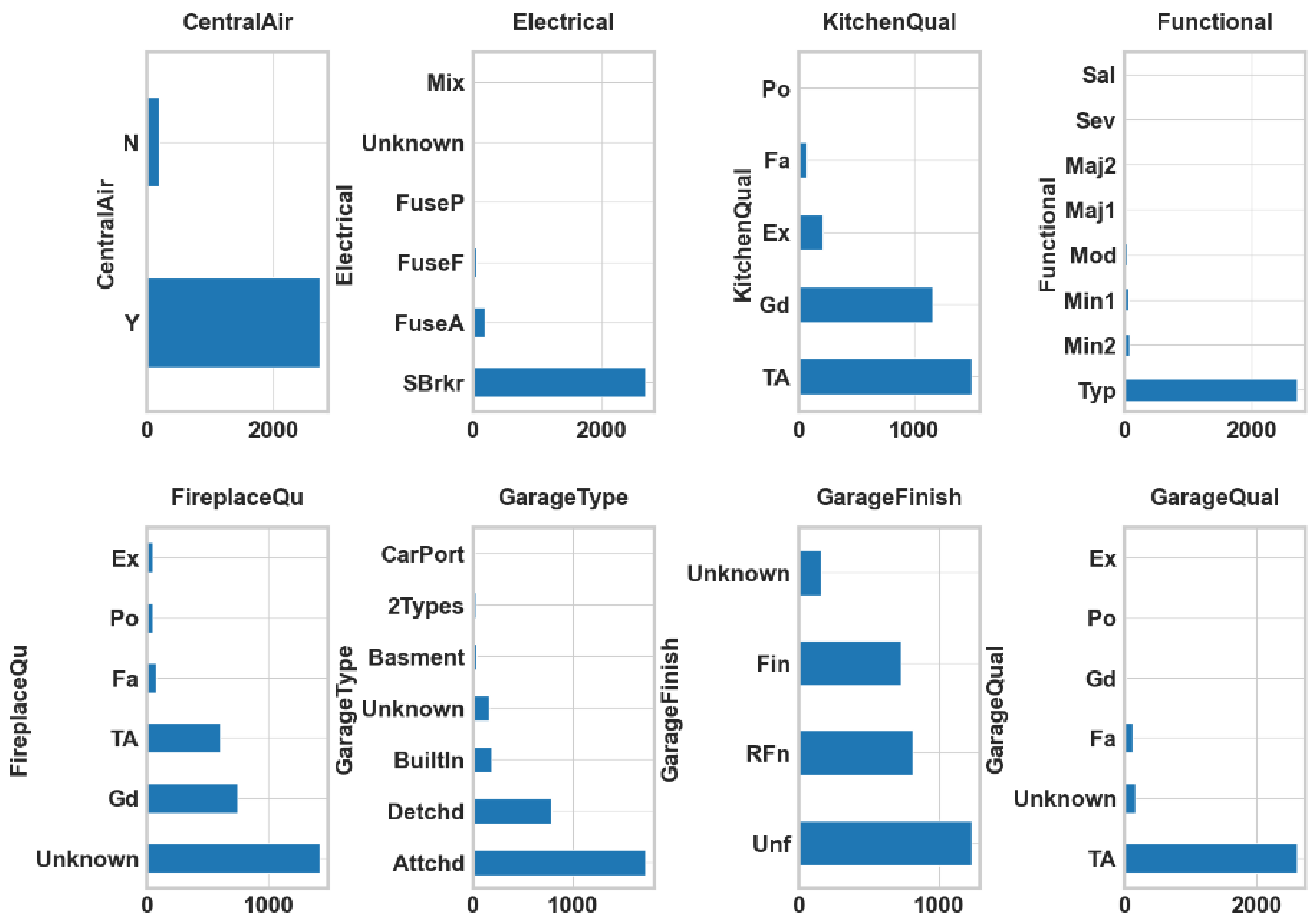
- **Fireplaces:** Number of fireplaces
- **FireplaceQu:** Fireplace quality
- **Garage Cars:** Size of garage in car capacity
- **Garage Area:** Size of garage in square feet
- 64 other features

Property Recommendation

Sale Price Prediction

Missing data were labeled ‘unknown’ for categorical features



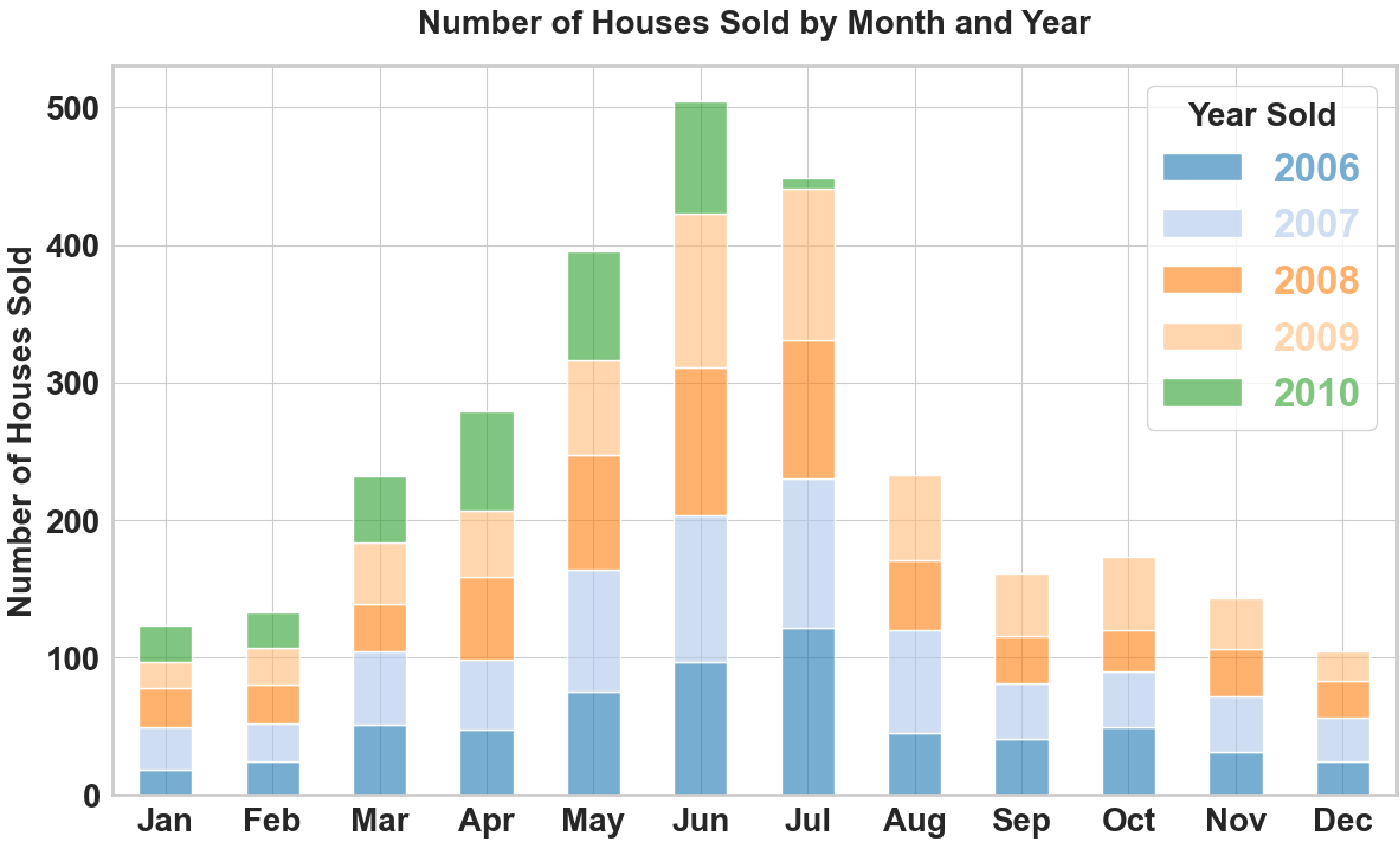


If a categorical column had more than 10% missing values, a new category called 'Unknown' was created. For columns with less than 10% missing values, they were imputed with the most frequent category.

House prices have linear relationship with area and quality

- The house of price increases with quality and area of the house
- The rate of change of house price is steeper for total basement area
- There are price ranges for same quality and area stemming a fan like structure more pronounced in 'AboveGround Living Area' and 'First Floor Area'
- There are some outliers in area features

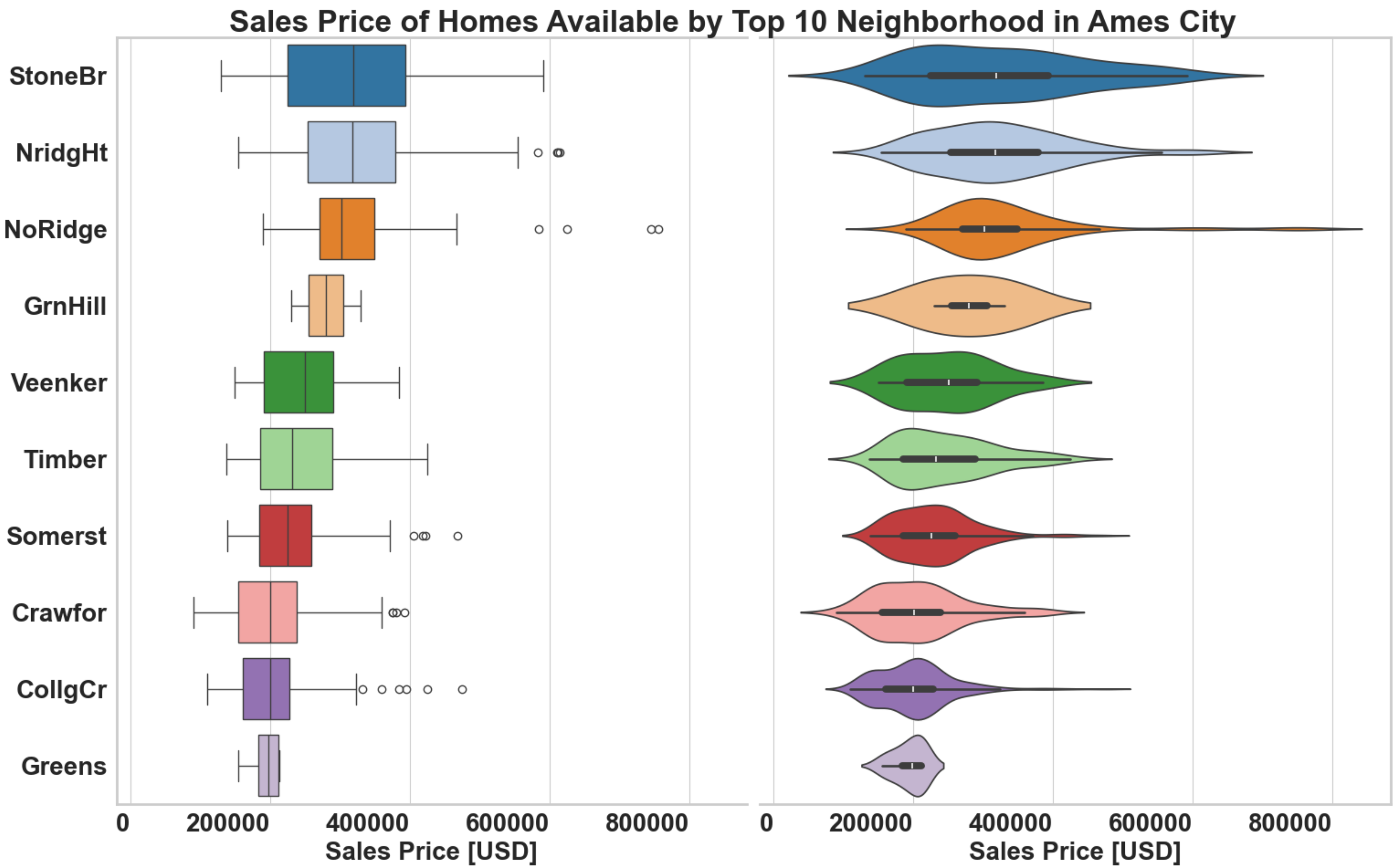
Q2/Q3: Peak season for home sales



House Sales Seasonal Trend

- Sales start increasing in May and peak in June
- Lowest transactions occur at the end of the year and extend into the first two months of the following year

Neighborhood-based median house prices



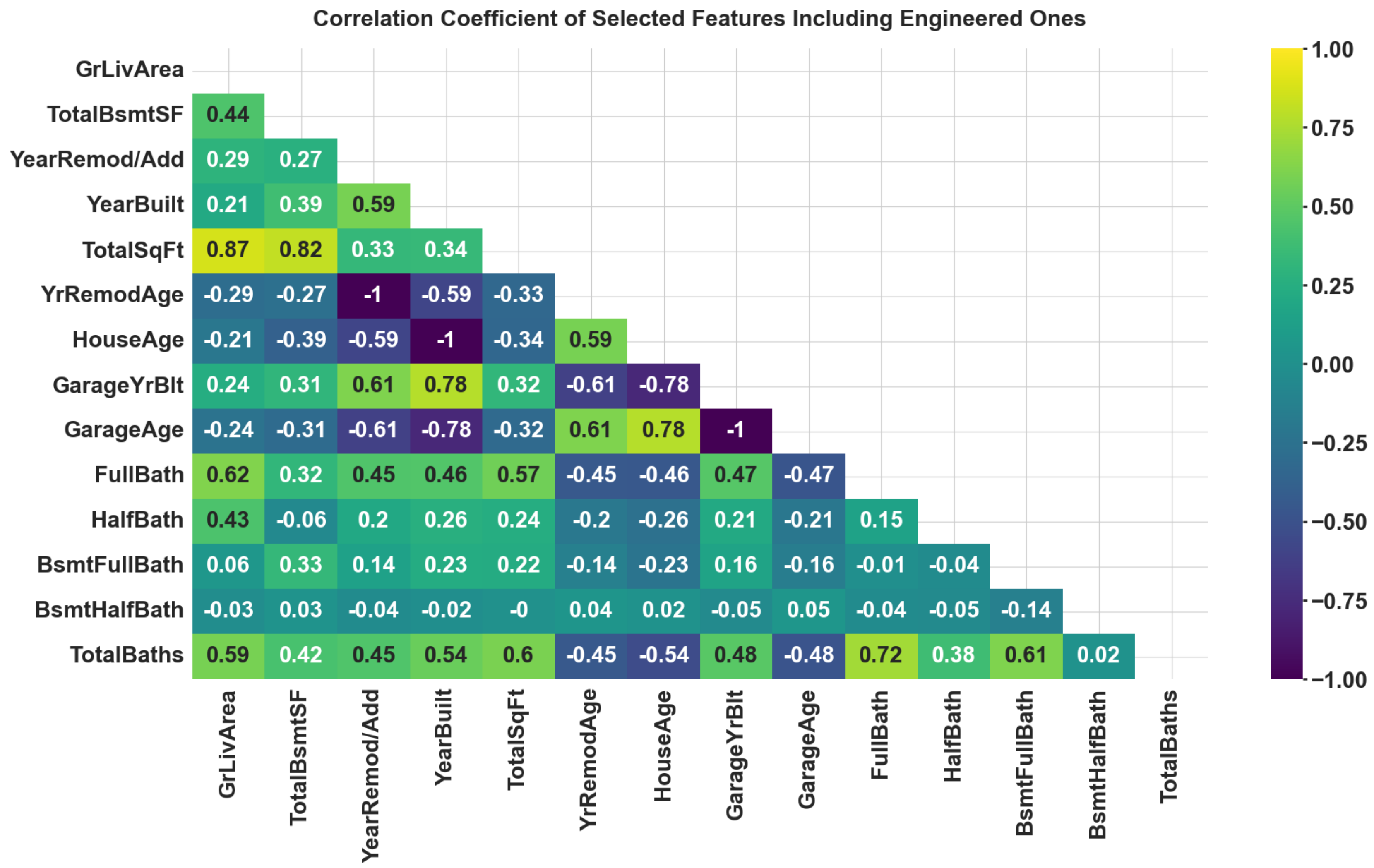
- House prices in some neighborhoods start from as low as $200K, while the highest-priced neighborhoods can reach up to $600K
- Median house prices vary depending on the neighborhood
- Certain neighborhoods exhibit outlier house prices, with some properties significantly deviating from the median range

Engineered features are highly correlated with original features

Engineered Features

- TotalBaths
- HouseAge
- YearRemodAge
- TotalSqFt

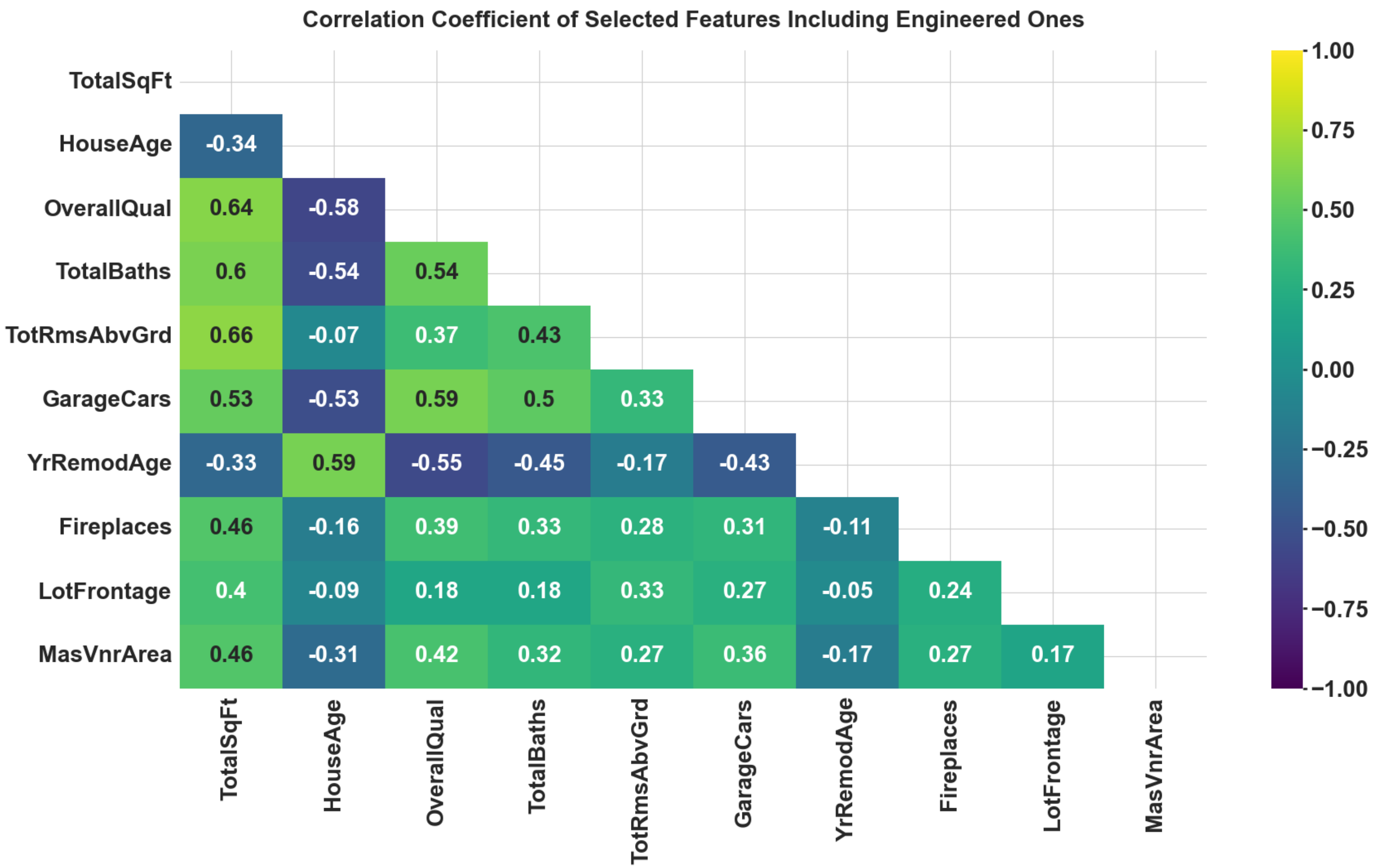
Features	VIF
TotalSqFt	3201.15
BsmtFinSF1	1021.55
BsmtUnfSF	954.11
2ndFlrSF	915.30
1stFlrSF	755.71
BsmtFinSF2	145.64
LowQualFinSF	11.83
GarageCars	5.60
GarageArea	5.44
TotRmsAbvGrd	4.50
HouseAge	4.49
GarageAge	3.20
TotalBaths	3.04
YrRemodAge	2.37



The top 10 numerical features were selected for modeling, ensuring the most impactful variables are used for accurate predictions

Feature	F-Score	P-Value
OverallQual	4342.9792	0.000000
TotalSqFt	3873.3856	0.000000
GarageCars	1752.9074	0.000000
TotalBaths	1709.5729	0.000000
GarageArea	1665.5998	0.000000
1stFlrSF	1519.5015	0.000000
HouseAge	1053.7745	0.000000
YrRemodAge	939.3147	0.000000
GarageAge	817.7759	0.000000
MasVnrArea	806.9772	0.000000
TotRmsAbvGrd	736.0766	0.000000
Fireplaces	718.7684	0.000000
BsmtFinSF1	543.6743	0.000000
WoodDeckSF	317.1991	0.000000
LotFrontage	280.3683	0.000000

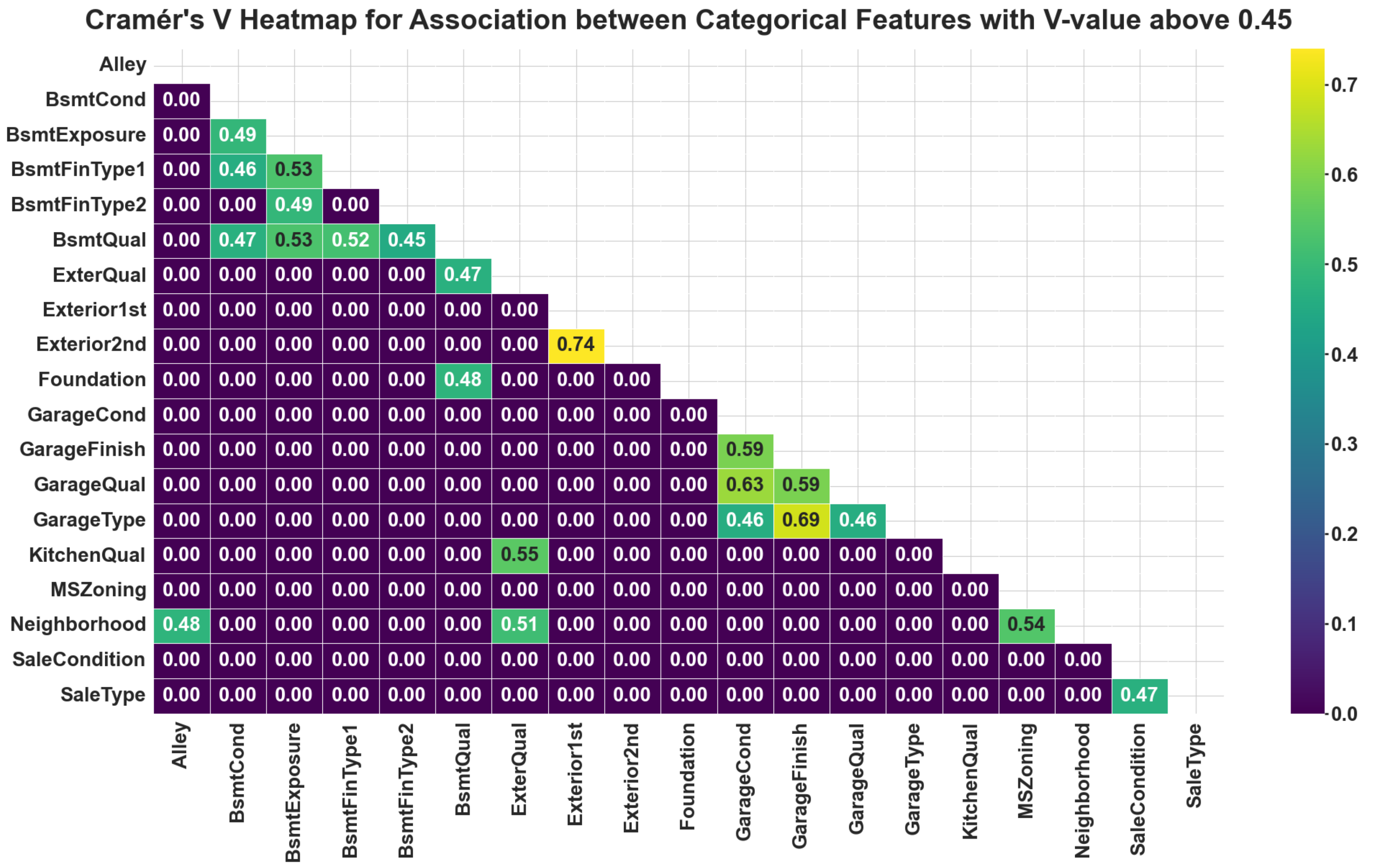
Variable	VIF
TotalSqFt	3.55
HouseAge	3.42
OverallQual	2.77
TotalBaths	2.10
YrRemodAge	1.94
TotRmsAbvGrd	1.93
GarageCars	1.90
Fireplaces	1.41
MasVnrArea	1.36
LotFrontage	1.36



The top 4 categorical features were selected using ANOVA and Cramer's V

ANOVA Test

Categorical Column	F-value	P-value
ExterQual	808.16	0
KitchenQual	555.20	0
BsmtQual	489.49	0
GarageFinish	341.35	1.30E-185
FireplaceQu	220.99	1.81E-195
CentralAir	194.12	1.52E-42
Foundation	186.77	3.43E-169
HeatingQC	161.99	2.40E-123
MasVnrType	133.61	1.70E-103
GarageType	130.70	4.59E-144
BsmtExposure	130.08	5.58E-101
BsmtFinType1	117.58	5.55E-131
Neighborhood	117.23	0



Minimum viable product (MVP): CatBoost trained with all features

- TotalSqFt and OverallQual emerged as the most significant features, contributing over 40% to the model's predictive power

- Over 10 features individually contribute 1 - 2% to the model's predictions

- The CatBoost model, trained with default parameters and using all input features, achieved an average out-of-fold (OOF) R2 score of 9.125%

```
# Identify and fill NaNs in categorical columns
cat_features = [col for col in X_train_new.columns if X_train_new[col].dtype == 'object']
# Define and train the default CatBoost Model
base_model = CatBoostRegressor(cat_features = cat_features, random_state = 42, verbose = 0)
base_scores = cross_val_score(base_model, X_train_new, y_train, cv = 5, scoring = 'r2')
print(f"Average r2 score for default CatBoost: {base_scores.mean():.4f}")
```

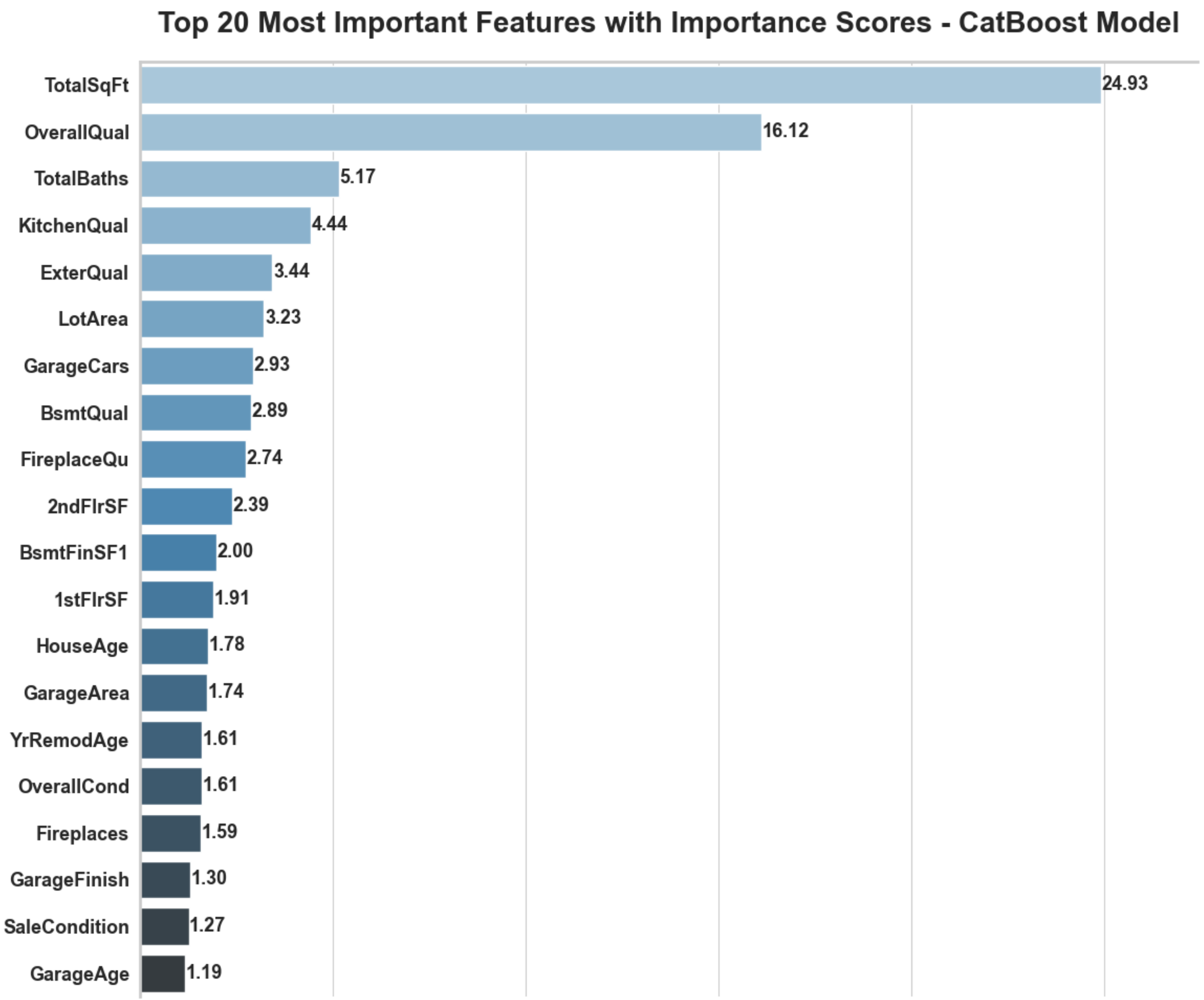
[131]

```
Average r2 score for default CatBoost: 0.9125
```

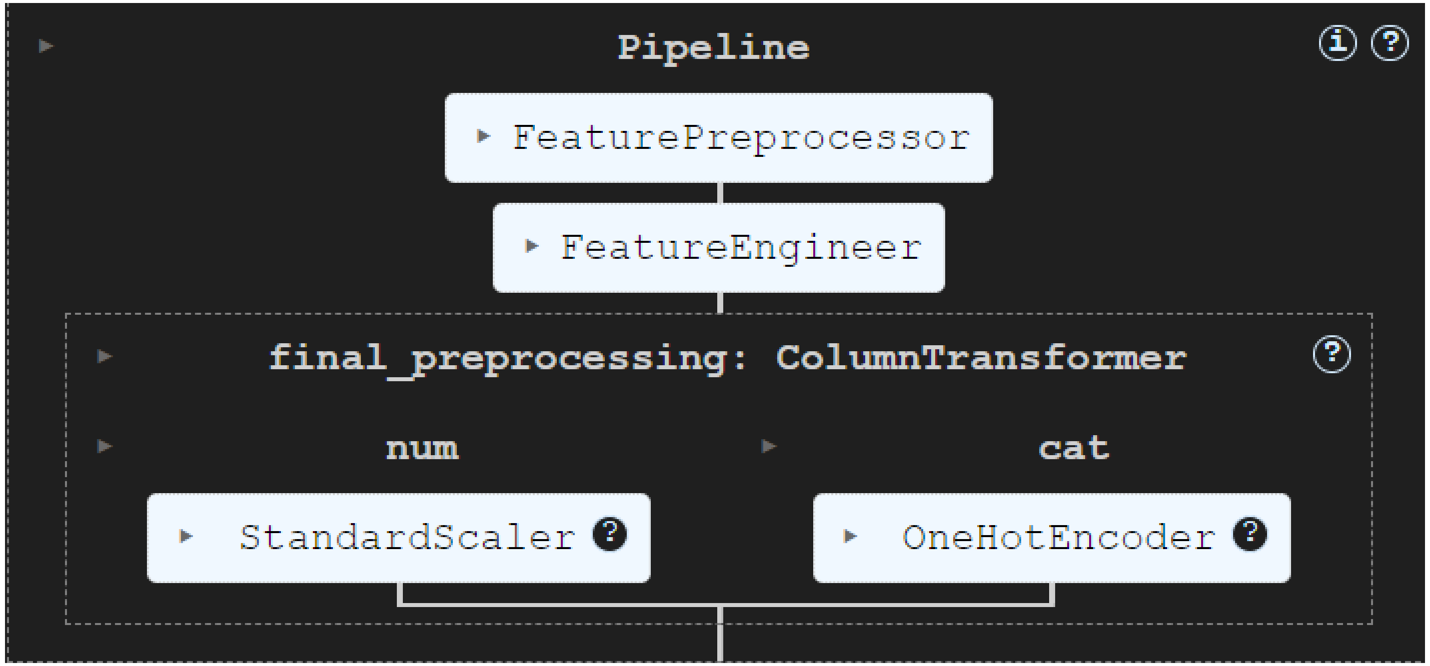
```
base_scores
```

[132]

```
array([0.88449204, 0.92523497, 0.92133909, 0.89634955, 0.93528412])
```



Automated preprocessing pipeline




```
Pipeline(steps=[('initial_preprocessing',                FeaturePreprocessor(categorical_features=['Neighborhood',
'FireplaceQu',                                              'KitchenQual',
'BsmtExposure'],                            numeric_features=['YearRemodAdd', 'YrSold',
'Fireplaces',                                           'LotFrontage',                                            'GarageCars',
'MasVnrArea',                                             'BsmtFullBath',                                            'GrLivArea',
'BsmtHalfBath',                                           'YearBuilt',                                             'OverallQual',
'TotalBsmtSF',                                           'HalfBath', 'FullBath',                                          'TotRmsAb...
Pipeline(steps=[('scaler',                                               StandardScaler())]),
['OverallQual',                                         'TotRmsAbvGrd', 'GarageCars',
'Fireplaces', 'LotFrontage',                                     'MasVnrArea', 'TotalSqFt',
'HouseAge', 'TotalBaths',                                      'YrRemodAge']),                                  ('cat',
Pipeline(steps=[('onehot',                                                OneHotEncoder(drop='first',
handle_unknown='ignore',                                                       sparse_output=False))]),
['Neighborhood',                                          'FireplaceQu', 'KitchenQual',
'BsmtExposure'])]))])
```

Feature Preprocessor

- Data loading, column names normalization and data validation
- Handles missing values using median imputation for numeric features and 'Unknown' for categorical features
- Consolidates rare categories based on threshold (merges categories < 8% into 'Other')

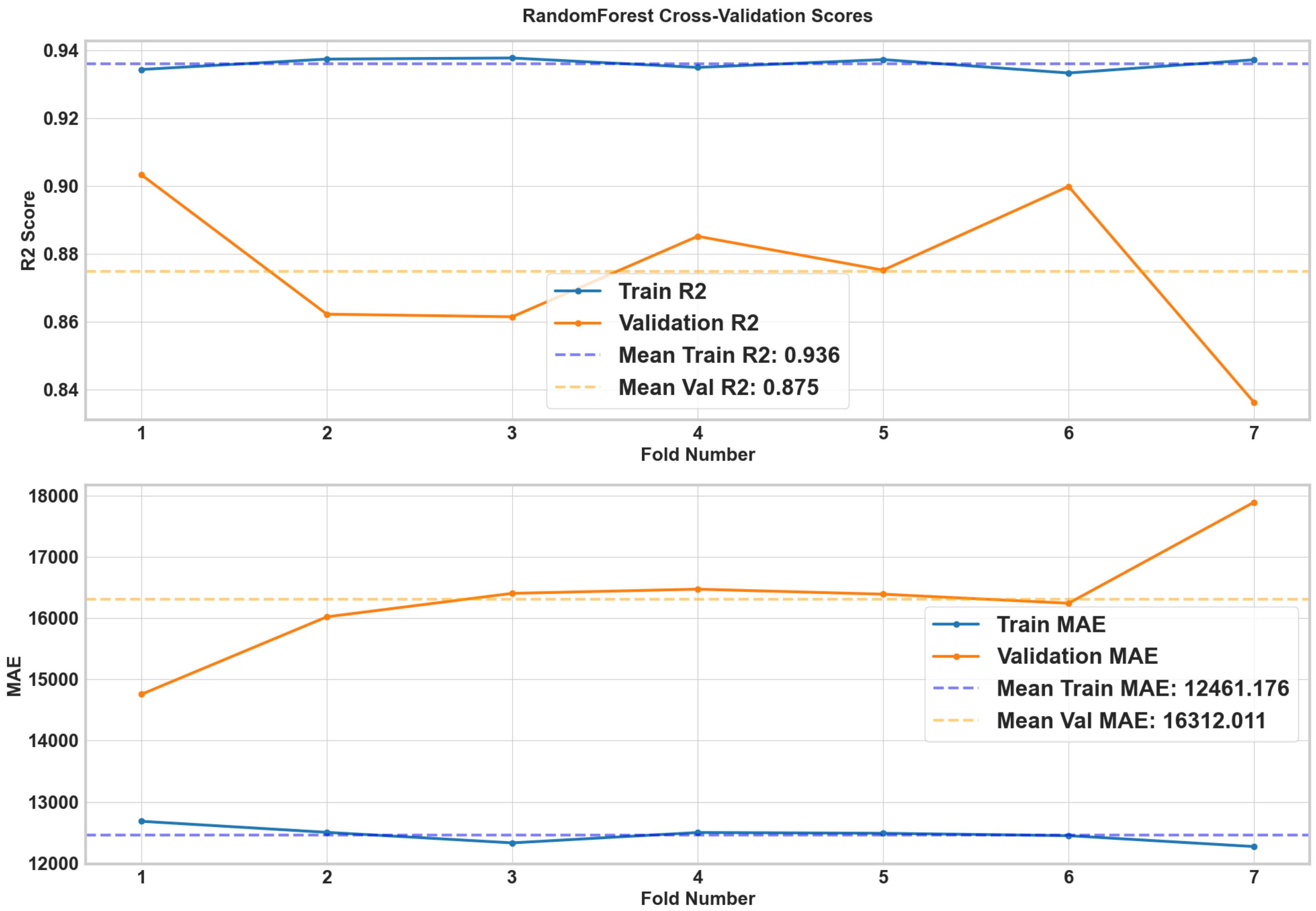
Feature Engineer

- Creates engineered features: TotalSqFt, HouseAge , TotalBaths, and YrRemodAge
- Automatically drops original features after engineering new ones

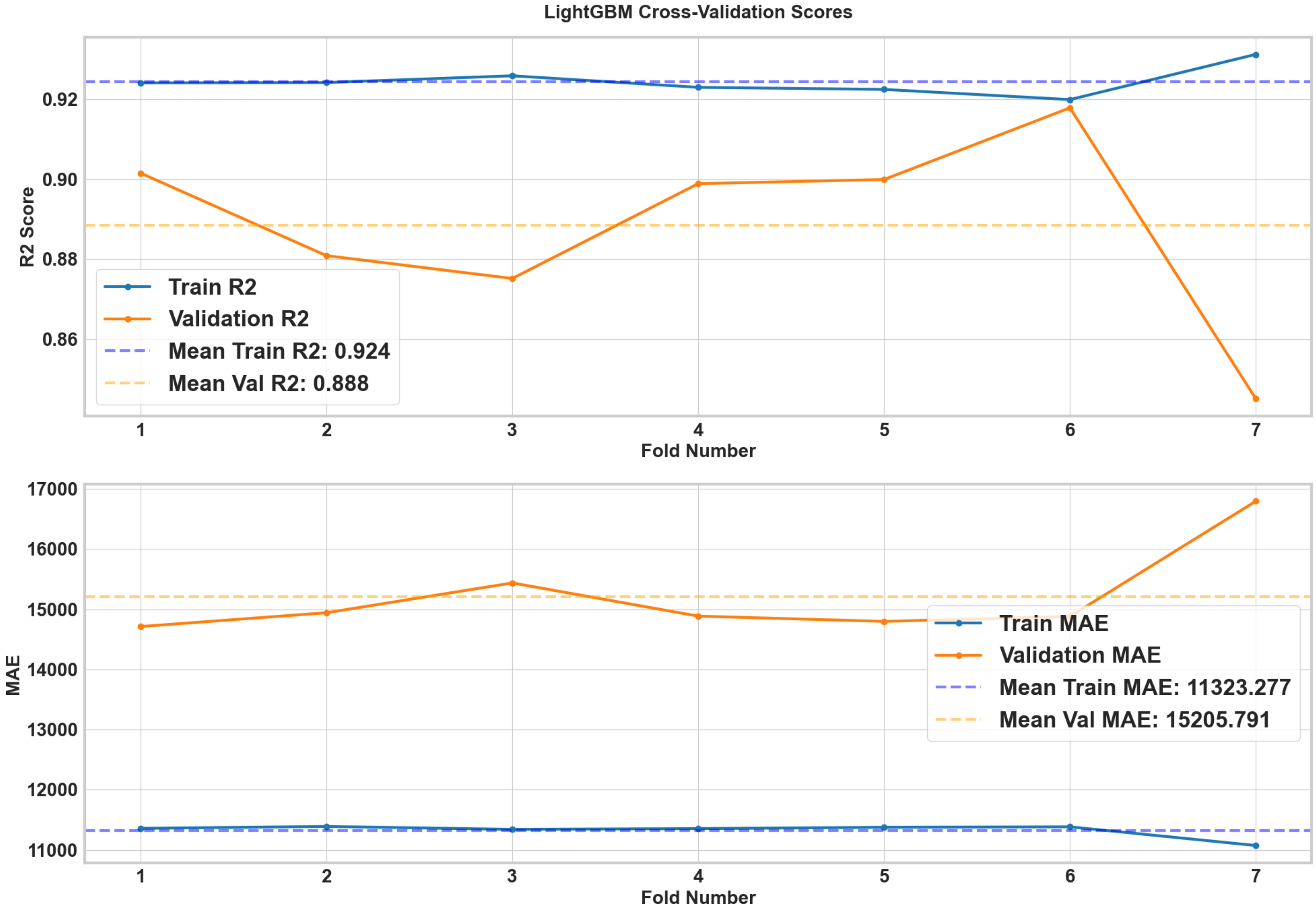
Final Column Transformers

- Numeric Pipeline: Applies StandardScaler to normalize all numeric features
- Categorical Pipeline: Uses OneHotEncoder with drop = 'first' and handle_unknown = 'ignore' for categorical variables

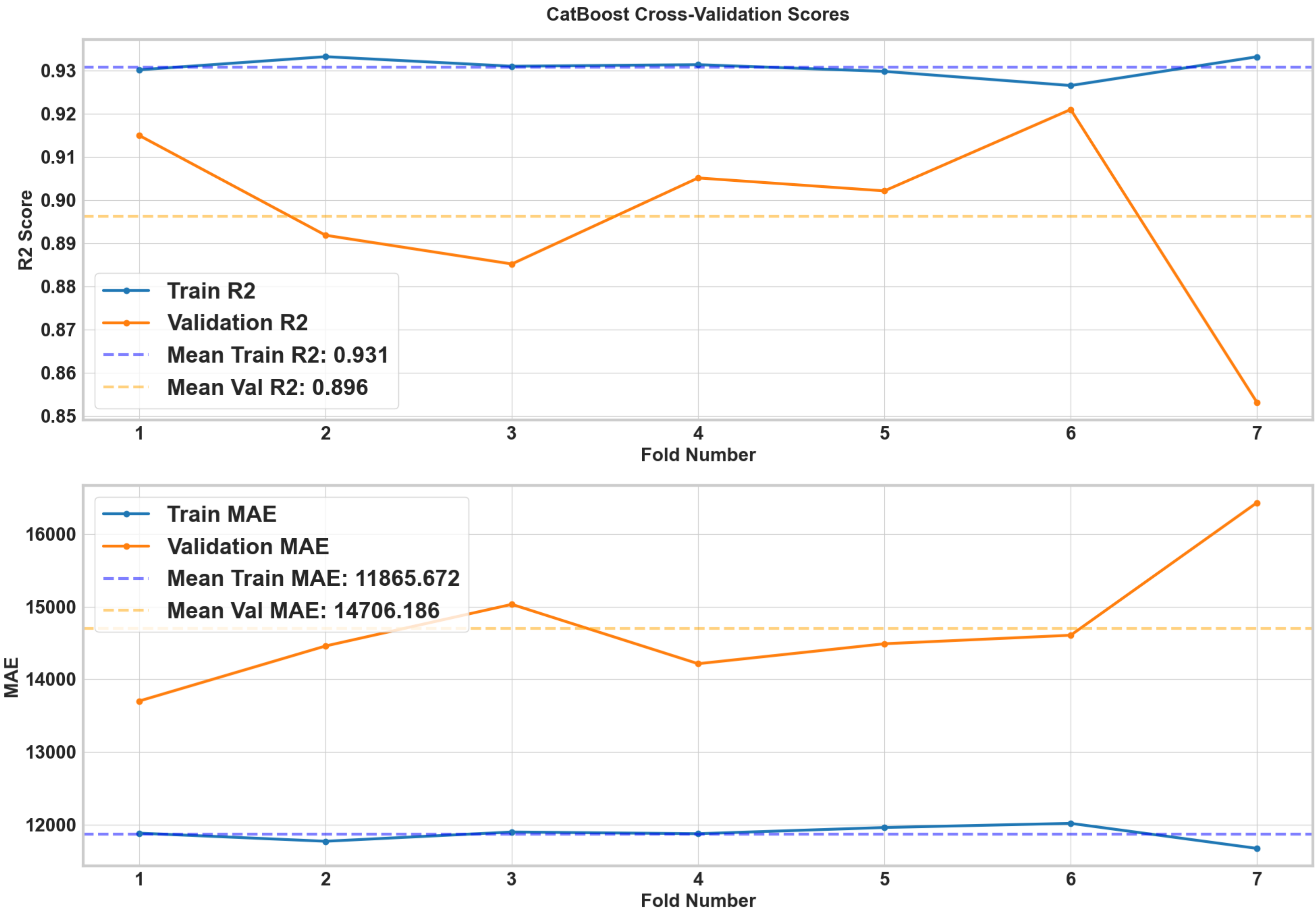
Cross validation shows slight variation in OOF prediction



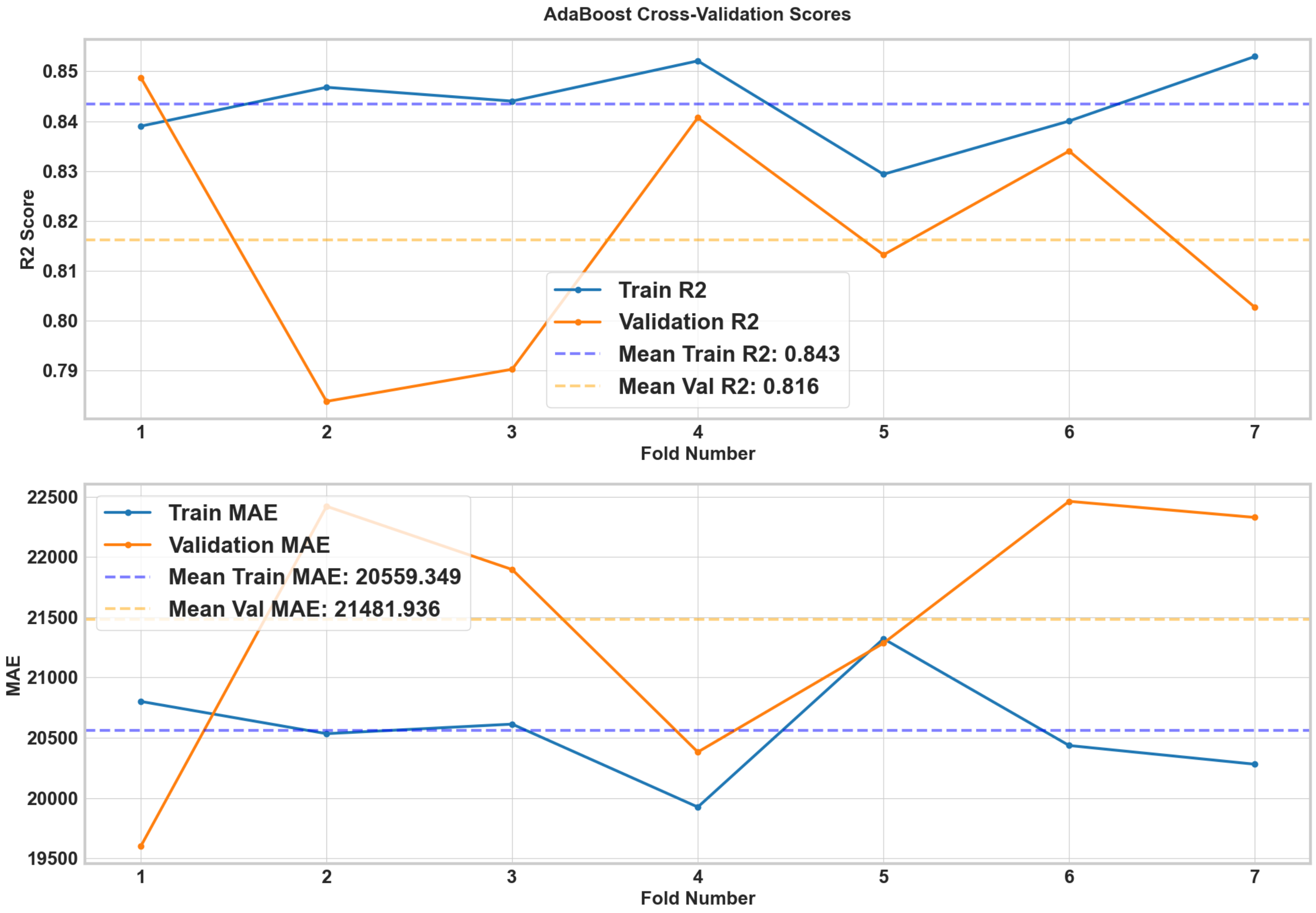
Cross validation shows slight variation in OOF prediction



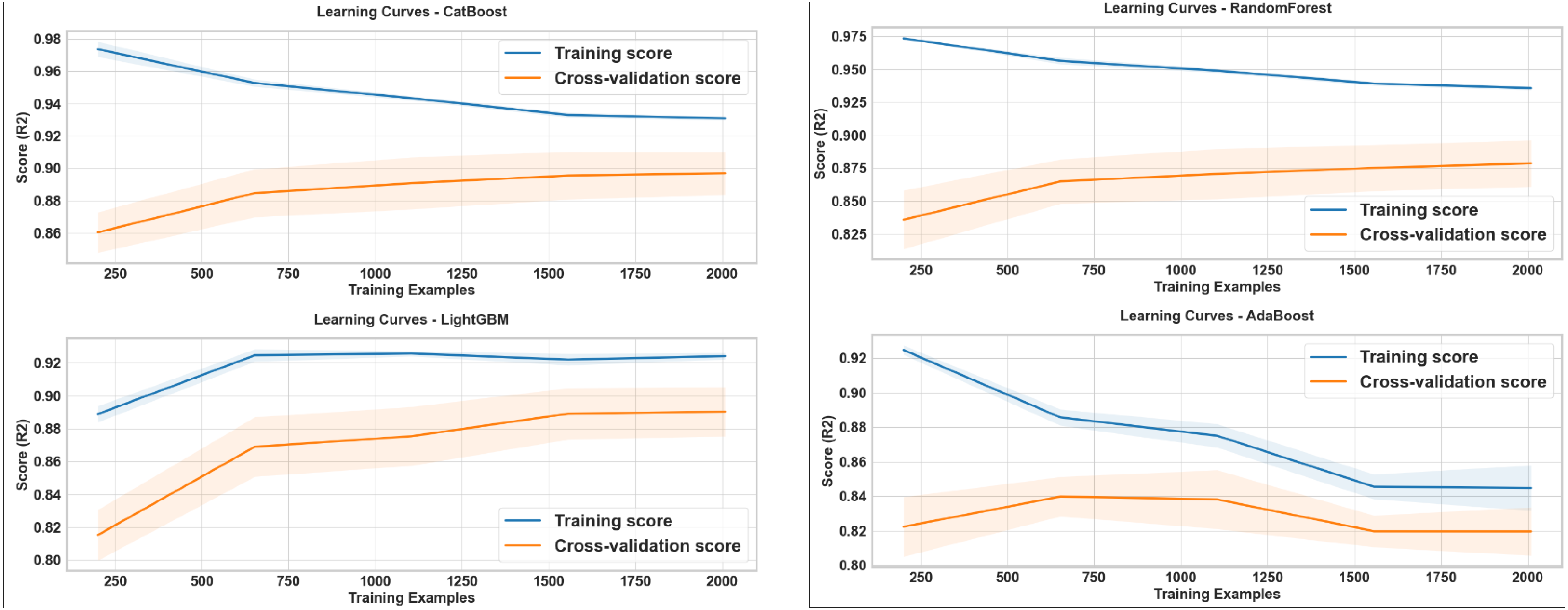
Cross validation shows slight variation in OOF prediction



Cross validation shows slight variation in OOF prediction



All models stabilize for sample sizes above 1500, but they consistently overfit



LightGBM excels in both predictive power and latency; CatBoost offers slightly enhanced predictive capabilities

Hyperparameter Tuned Model Performance

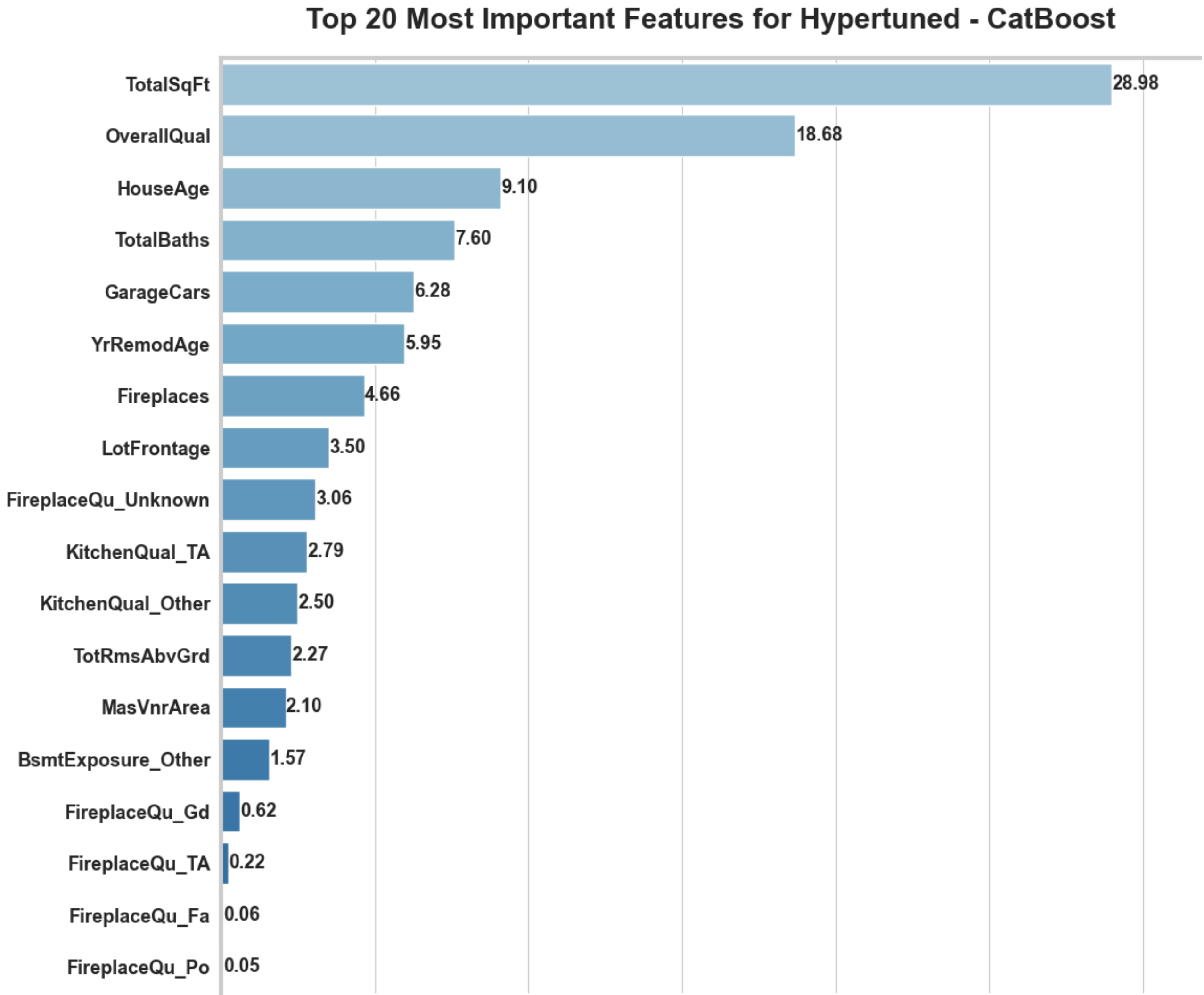
Model	Train R2	Test R2	Train MAE	Test MAE	Train MAPE	Test MAPE	Training Time
CatBoost	0.9362	0.9386	11091.4807	12893.8678	0.075036	0.080996	2.41s
LightGBM	0.9371	0.9280	9783.1770	13702.3802	0.067831	0.087694	0.12s
RandomForest	0.9540	0.9242	10225.5118	14146.3613	0.067312	0.089740	1.70s
AdaBoost	0.8378	0.8526	20584.1489	20969.9087	0.133344	0.125758	0.21s
Ensemble	0.9364	0.9295	12071.4640	14215.5670	NaN	NaN	N/A

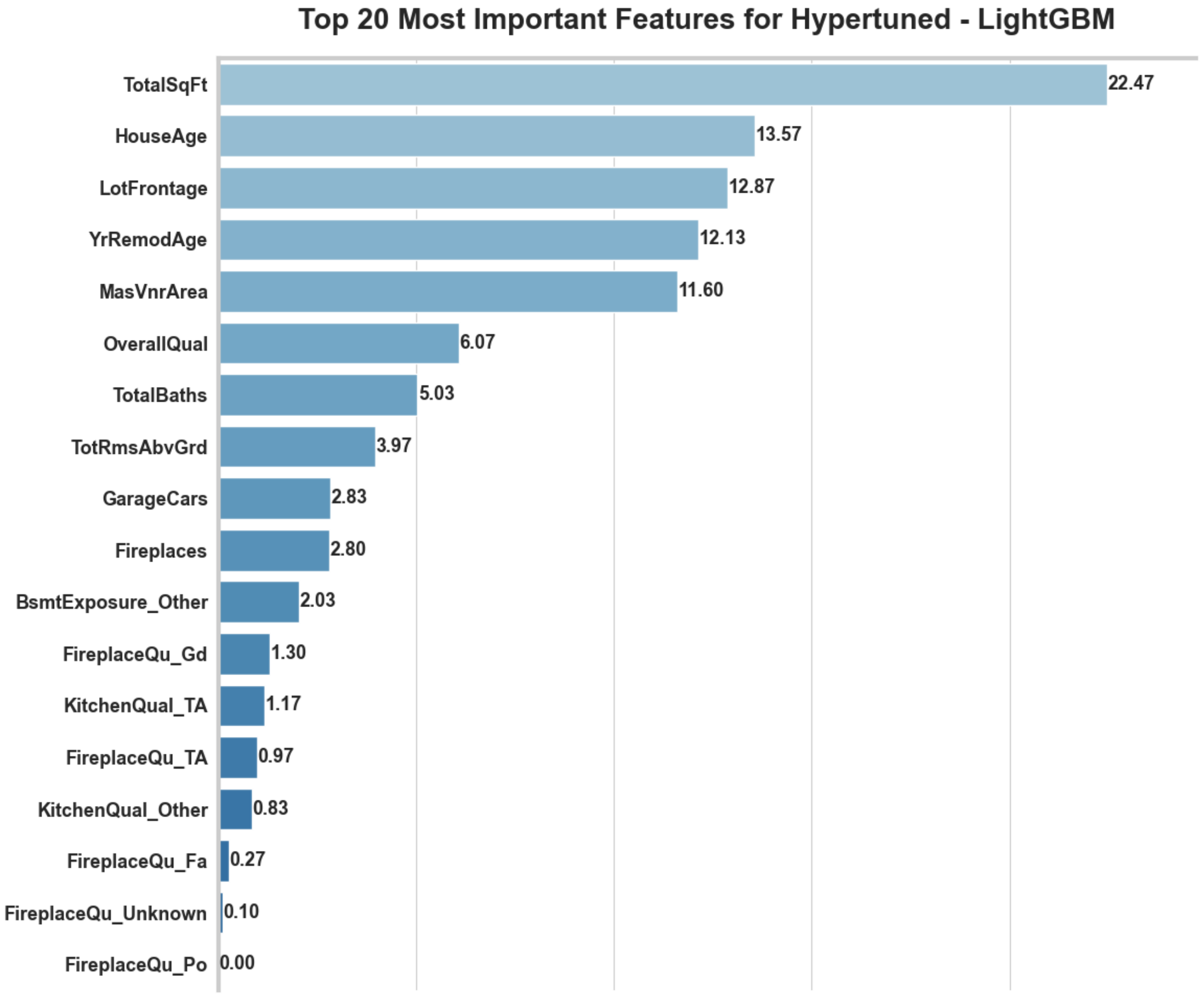
```
best_params = {
  "CatBoost": {
    "colsample_bylevel": 1.0,
    "depth": 5,
    "early_stopping_rounds": 500,
    "eval_metric": "MAE",
    "iterations": 1000,
    "learning_rate": 0.03,
    "min_data_in_leaf": 1,
    "objective": "MAE",
    "subsample": 0.9,
    "verbose": 0
  },
  "LightGBM": {
    "colsample_bytree": 0.9,
    "learning_rate": 0.15,
    "metric": "mae",
    "min_child_samples": 25,
    "min_child_weight": 0.001,
    "n_estimators": 100,
    "num_leaves": 31,
    "objective": "regression_l1",
    "reg_alpha": 0.0,
    "reg_lambda": 0.0,
    "stopping_rounds": 50,
    "subsample": 0.9
  },
  "RandomForest": {
    "max_depth": 9,
    "min_samples_leaf": 2,
    "min_samples_split": 5,
    "n_estimators": 150
  },
  "AdaBoost": {
    "learning_rate": 0.8,
    "loss": "linear",
    "n_estimators": 50
  }
}
```

Latency vs Performance Trade-off

- LightGBM emerges as the most efficient model with just 0.12s training time while maintaining strong performance (R^2 0.928, MAE 13702)
- CatBoost achieves the best test accuracy (R^2 0.939) but requires 20x longer training time (2.41s) than LightGBM
- RandomForest's higher training R^2 (0.954) comes at the cost of longer training time (1.70s), suggesting potential overfitting
- The choice between CatBoost and LightGBM would depend on whether the 1% improvement in R^2 justifies the 20x increase in training time

Feature importance: How different models tell different stories

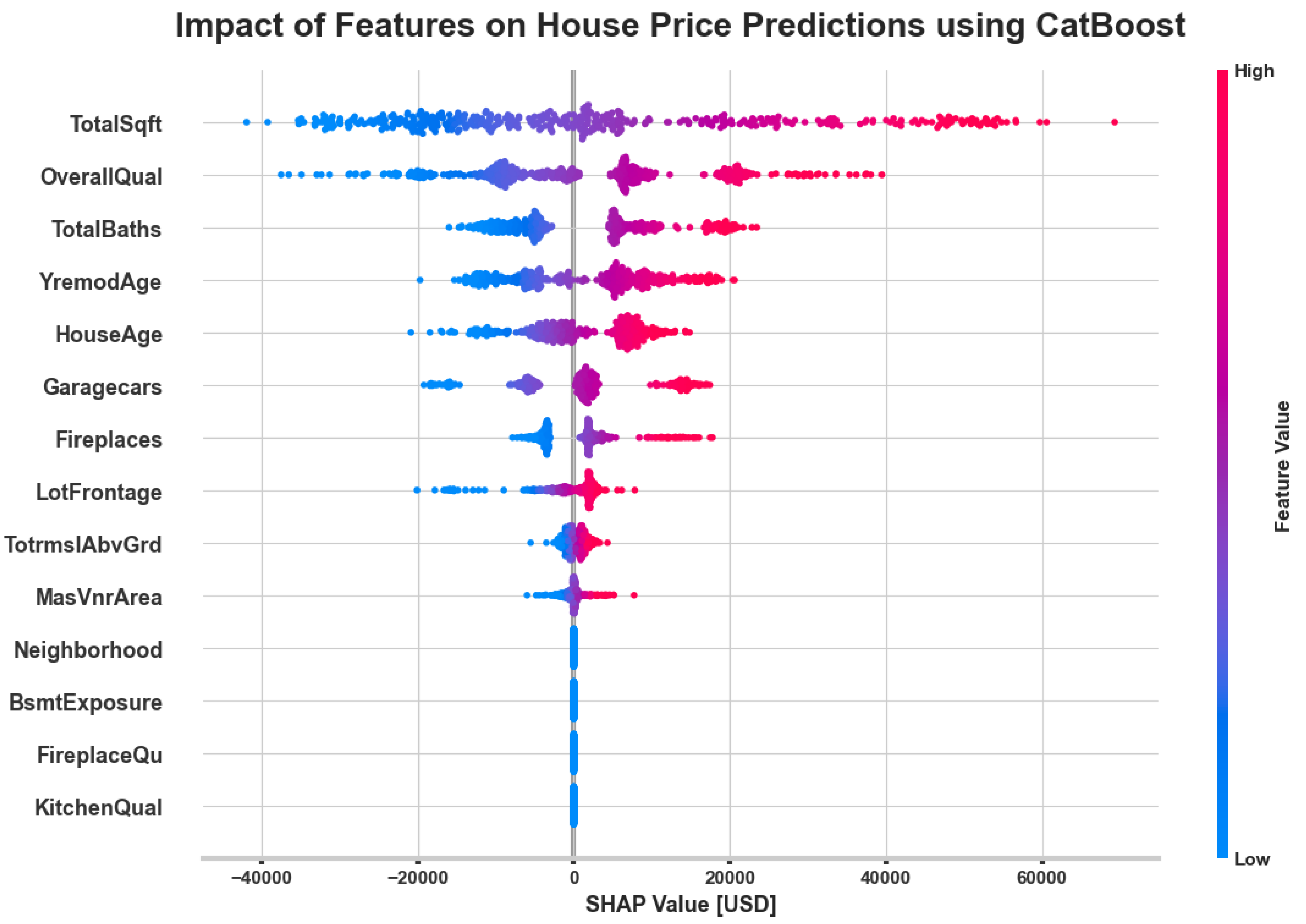




CatBoost emphasizes **quality metrics**: OverallQual, TotalBaths, and KitchenQual rank higher. **LightGBM** gives more weight to **physical attributes**: LotFrontage and MasVnrArea show much higher importance

$$ value impact: What each home feature adds

- Total Square Footage (TotalSqft) has the largest impact range (~60k USD) and shows consistently positive influence for larger values
- Overall Quality (OverallQual) is the second most influential feature, with higher quality scores strongly driving up prices
- Neighborhood and quality-related categorical features (KitchenQual, FireplaceQu, BsmtExposure) show clustered impacts



Potential Directions

- **LLMs in Proptech:** Utilizing LLMs to scan and extract valuable information from property and legal documents, enhancing property transaction efficiency.
- **Foreclosure Predictions:** Implementing AI models to forecast foreclosure risks and assess buyer preparedness after listing a property for sale.
- **Image Analysis for Property Assessment:** Using AI for image analysis to detect property damages, aiding in accurate property valuation.
- **Proactive Real Estate Services:** Collaborating with banks and mortgage lenders to create databases tracking mortgage defaults, offering proactive property management.
- **AI/ML as SaaS:** Providing AI and ML technologies as a service to the real estate sector, enabling unprecedented insights and operational efficiency.

Acknowledgements

I would like to extend my heartfelt gratitude to everyone who provided invaluable assistance during this project.

- ❖ Vivian S. Zhang
- ❖ Cole Ingraham
- ❖ My cohorts: James Seykot, Margaret Bowers, Oreste Rukundo, Amiyo Chattarjee