King County Prediction Report

Dataset: House Sales in King County

Introduction

Our chosen dataset provides information about homes sold in King County, USA between May 2014 and May 2015. The dataset provides information about the attributes of a home (size, # of rooms, quality of construction/design), its location (zipcode, view, latitude/longitude), and neighborhood information, alongside the price at which the home was sold for. Each row of data contains 21 unique columns of information, where the features are:

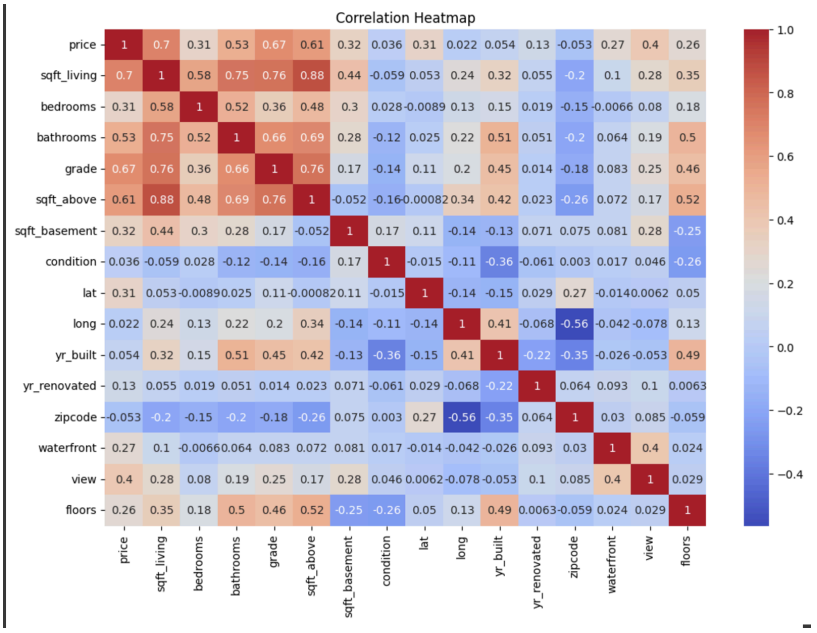
1. id
2. date (of the home sale)
3. price (in dollars) of the home sale
4. # of bedrooms
5. # of bathrooms
6. Sqft_living (square footage of the interior living space)
7. Sqft_lot (square footage of the land space)
8. # of floors
9. waterfront - (a binary feature that = 1 if it is on a waterfront and 0 if it isn't)
10. view - (An index from 0 to 4 of how good the view of the property was)
11. condition - (An index from 1 to 5 on the condition of the apartment)
12. grade - (An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design)
13. sqft_above
14. sqft_basement
15. yr_built
16. yr_renovated
17. zipcode
18. latitude
19. longitude
20. sqft_living15 (sqft_living for the nearest 15 neighbors)
21. sqft_lot15 (sqft_lot for the nearest 15 neighbors

There are a total of 21613 rows with no missing values (all rows have their 21 attributes filled out accordingly). Thus, all rows can be used for analysis and model development without further data cleaning. Given the broad range of features and large sample size, this dataset was already suitable for analysis, specifically for analyzing the prices of the house sales.

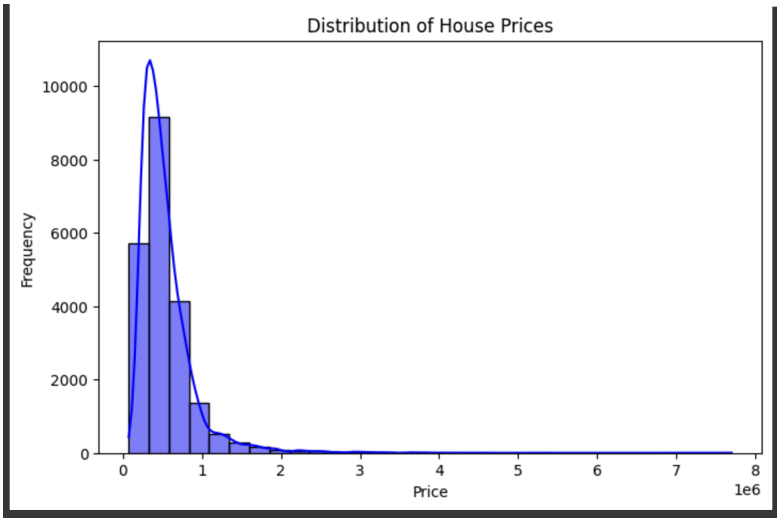
Exploratory Data Analysis

To first determine the validity and reliability of the data, we displayed basic summary statistics about each of the variables, including their associated means, medians, standard deviations, etc, where each major feature (price of house, grade) seemed to have few outliers. In addition, we aimed to get rid of

unnecessary attributes at this stage. “Sqft_living15” and “sqft_lot15” were both unnecessary as they were redundant, and the id and date of the sale would not matter for our models. Next, to get insight into the relationships between the different attributes, we created a correlation matrix that displated the correlation between two unique variables. From this heatmap, we saw that many variables were strongly correlated with price. For instance, sqft_living has a strong positive correlation with price (0.7), meaning larger living spaces often sell for more. Grade is also positively correlated with price (0.67), suggesting better quality houses sell at higher prices. Weak correlations, like between sqft_basement and price (0.32), indicate less influence on price. From this information, we determined that price would be a suitable regrassand, acting as the Y variable for our regression models.



After picking the dependent variable, we aimed to gain more information into the distribution of the price values in the dataset. From the histogram that we created, house prices seem to be skewed to the right, with many house sales falling below $1 million. The right-skewed shape indicates a few high-value homes as outliers, helping us understand the overall pricing landscape in King County at the time. The right-skewed distribution of house prices could pose challenges for predictive modeling, as regression models may struggle to accurately predict prices for most homes due to the influence of outliers—especially high-value properties. To avoid this potential issue, we decided to log transform our price variable for future model selection, allowing the skewness to not affect prediction significantly. Also, the average house price appeared to be around $540,000, with an average living space of 2,080 sqft and 1.5 baths per bed. This gave us good insight into the variables surrounding and impacting the prices that houses are sold for. Overall, from our exploratory data analysis, we were able to determine the feature that we wanted to predict, the importance of log transformation for price, as well as learn more about what attributes would be good predictors for price.



<u>Feature Selection</u>

Before creating our models, we first needed to decide what regressors to use to predict housing price. From the correlation heatmap above, we knew that sqft_living (sqft of the interior living space) and grade of the construction would be necessary, as they resulted in strong correlations with housing price. Although sqft_above and sqft_basement had a positive correlation, we knew that it would be redundant to add these to a model, as sqft_living was already a key regressor. Thus, as no other features had strong correlations, we decided to use feature engineering to create regressors that can better represent its relationship with price. One of these new features was "bath_bed_ratio", where we took the ratio of bathrooms to bedrooms in the house to make a regressor that more accurately represented all of the rooms in a given house (increased the correlated coefficient of bedrooms). This approach seemed to aid in better prediction accuracy (more information later). Also, we combined the "view" and "waterfront" features into one regressor dubbed "appeal", which increased the usability of the waterfront feature and made the regressor more clear as to how the location of a house can impact its price. Lastly, we added features that qualitatively made sense to predict price. The condition of the houses would positively influence the prices that they are sold for, as well as the location of the houses, where the inclusion of these attributes improved our models. Below is the final X and y test splits that were used to train our models.

```
features = ['sqft_living', 'grade', 'bath_bed_ratio', 'appeal', 'lat', 'condition']
X = kc_house_data[features]
y = np.log1p(kc_house_data['price'])  # Log-transform the target to reduce skewness

# Impute missing values with mean
imputer = SimpleImputer(strategy='mean')
X = imputer.fit_transform(X)

# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Model 1: Linear Regression

We first trained our X features on a basic linear regression model, utilizing quantitive results, such as root mean squared error and a r^2 coefficient, to evaluate our models. After training our model with X_train and acquiring y_pred after running the model on X_test, we got the following mean squared error and r^2 values.

```
Linear Regression:
RMSE: 0.3115813507980779
R2 Score: 0.7418958187360476
```

These values served as a good starting point for regression analysis, creating a model that accounted for 74% of the variance in housing price in the dataset, as well as a small distance between predicted and actual values. This information also gave us good insight into the baseline for testing with other features and models, and showed us the positive impact of the feature engineering that we did early. For example,

after replacing “bath_bed_ratio” with simply the number of bathrooms or bedrooms, we saw that the r^2 values decreased by a sizeable amount (0.68 vs. 0.74). Also, to test the validity of these statistics, we implemented 5-fold cross validation to determine if the R^2 value was accurate.We ran this cross-validation algorithm on the training dataset and evaluated the performance of the model on each fold. By averaging the R² scores across the five folds, we ensured that the model's performance was stable and not overly dependent on a specific subset of the data. This approach helped us verify the generalization ability of the model and reduce the risk of overfitting, ensuring more reliable predictions when applied to unseen data. Next, we wanted to evaluate how the inclusion of a L1 regularization value impacted the linear regression model.

Model 2: Lasso Regression

To first start our development, we needed to first determine the ideal alpha value of the L1 regularization term. To do this, we implemented a grid search cross validation algorithm that tuned hyperparameters of alpha (shown below) by fitting each L1 term to the overall model, and returning the alpha value that results in the best r^2 value. In our case, the best alpha value was 0.01, which we used to train our data on a Lasso regression model, resulting in the following RMSE and R^2. From these results, we see that the variance accounted for in the Lasso model is marginally lower than the original linear regression model, with a higher mean squared error. As the L1 term did not result in a better model fit, we decided to seek non-linear methods.

```python
# Define the range of alpha values to test
param_grid = {'alpha': [0.01, 0.1, 0.5, 1, 10, 50, 100]}
lasso_cv = GridSearchCV(Lasso(random_state=42), param_grid, cv=5, scoring='r2')
lasso_cv.fit(X_train, y_train)

# Best alpha value
best_alpha = lasso_cv.best_params_['alpha']
print("Best alpha:", best_alpha)
```

```
Lasso Regression:
RMSE: 0.3123133862823941
R2 Score: 0.7408327928377418
```

Model 3: Polynomial Regression

To create a polynomial regression model, we transformed the X data by a degree of 2. The decision to use a degree of 2 was based on the understanding that the relationship between the housing price and its regressors still exhibited a general linear trend, but might benefit from the introduction of some curvature, particularly in the more complex interactions between features like sqft_living, grade, and condition. By choosing a relatively low degree, we aimed to avoid overfitting, which can occur when models become too complex relative to the amount of available data. After following the same procedures and getting an RMSE = 0.29 and R^2 = 0.77, we noticed that a non-linear model might be more appropriate for the relationship between the housing features and its price, as the root mean square error decreased and the accounted variance increased. The inclusion of quadratic terms helped

```python
from sklearn.model_selection import cross_val_score

# Perform cross-validation (5-fold) on the training data
cv_scores = cross_val_score(poly_model, X_train_poly, y_train, cv=5, scoring='r2')

print(f"Cross-Validation R² Scores: {cv_scores}")
print(f"Mean Cross-Validation R²: {cv_scores.mean()}")
```

```
Cross-Validation R² Scores: [0.76083992 0.76289455 0.76900117 0.76372752 0.75255126]
Mean Cross-Validation R²: 0.7618028826901535
```

account for more of the complexity in how various features interact and affect the pricing. However, we had to make sure that these results were statistically significant and weren't a result of overfitting. Thus, we followed the same 5-fold cross validation procedure we did for linear regression, calculating the mean r^2 and saw that it was very similar to 0.77, validating our results. Next, we aimed to obtain better results through another non-linear supervised learning procedure.

Model 4: Random Forest

Lastly, we ran our data through a random forest regressor with a random state equal to 42, obtaining a RMSE=0.244 and R^2 value=0.83. The random forest resulted in the best accuracy and reliability for price prediction, strongly indicating that the dataset would be better represented in a non-linear model. We also took a look into the importance of each feature for the random forest model, where we see that the grade of construction and latitude of the house have the strongest involvement in predicting the housing price. The condition of the house seems negligible in this model, indicating that grade and condition are not mutually exclusive and grade might already capture much of the variance typically associated with the condition of the house. These findings support the idea that a more sophisticated model like random forest can reveal deeper insights into the relationships between features that might not be as apparent in simpler linear models. In addition, we also trained our random forest with gradient boosting (XGBoost), improving our r^2 score to 0.842 and minimizing error.

```
sqft_living: 0.222244095898631 64
grade: 0.35355184373251963
bath_bed_ratio: 0.031774531317784356
appeal: 0.027808874237526187
lat: 0.3477283056764787
condition: 0.01689234913705956
```

Other Potential Regressors

Below are various features transformed from the dataset that we attempted to add to the model, but ended up in lowering the accuracy of prediction. We found that "price_per_sqft" caused massive overfitting, as it was derived directly from the price itself. Also, attempting to combine all the house features and its physical attributes didn't significantly affect its association with price, where all it did was simply increase the root mean square error for price prediction. We also attempted to use features already in the dataset, such as the number of bathrooms, the zipcode as a categorical variable, and longitude of the house, but found insufficient evidence for its inclusion in the model.

```
kc_house_data['price_per_sqft'] = kc_house_data['price'] / kc_house_data['sqft_living']
kc_house_data['total_rooms'] = kc_house_data['bedrooms'] + kc_house_data['bathrooms']
kc_house_data['age'] = 2024 - kc_house_data['yr_built']
kc_house_data['home_features'] = kc_house_data['floors'] + kc_house_data['bathrooms'] + kc_house_data['bedrooms']
```

Overall Qualitative Results/Future Directions

From the model fits, we can see that the housing prices are heavily related to the quality level of the construction and design of the house (grade), the latitude of where the house is located, and the overall size of the living space in the house through a non-linear relationship. We can also see that there is some influence for the number of rooms of the house, as well as its scenic surroundings. This data suggests that houses located closer to urban areas (as indicated by higher latitude values) tend to have higher prices, likely due to proximity to amenities, work opportunities, and desirable features of urban living. Additionally, the high importance of grade further emphasizes the role that construction quality and design play in determining housing price, with higher-quality homes commanding a premium in the market.

For future predictions, we can incorporate additional features such as the proximity to schools, parks, and public transportation to refine our understanding of how location factors into pricing. We could also experiment with adding interaction terms between variables (e.g., sqft_living and latitude) to capture more nuanced relationships between size and location, validating the idea that the data does not follow a linear model. Furthermore, exploring more advanced machine learning techniques, such as gradient boosting or neural networks, could help further improve predictive accuracy, particularly in handling more complex, non-linear patterns. Another direction could be analyze how home prices have evolved over time, as this dataset only captures data from 2014-2015. As the housing market is very unstable and dynamic, a larger timeframe could provide more accurate price predictions, especially in rapidly changing real estate markets.