1. *Introduction*

The price of a house depends on numerous factors, including its location, size, number of bedrooms and bathrooms, proximity to a waterfront, elevation, and many more. These varying factors make it challenging for stakeholders such as real estate agents, brokers, buyers, sellers, and academic researchers to accurately estimate house prices.

Predictive analytics is a powerful tool that can help address this issue. While it may not provide a perfect solution, it makes data-driven decision-making possible in critical areas like buying and selling homes, targeting the right audience, estimating market trends, and much more.

In this paper, we discuss how machine learning techniques can be used for data mining and house price prediction. Our dataset, originally sourced from Kaggle, is titled “House Sales in King County, USA.” It contains house sale prices for King County, including Seattle, covering transactions from May 2014 to May 2015.

Our research falls within the real estate domain and can be valuable to students, researchers, academia, real estate businesses, and industry stakeholders.

1. *Dataset Overview*

The dataset used for this analysis consists of 21,613 records, each containing 21 features. These features capture key residential property characteristics.

Following is the description of all 21 features:

| **Feature** | **Description** |
| --- | --- |
| Price | Sale price of the house. |
| Bedrooms | Number of bedrooms. |
| Bathrooms | Number of bathrooms. |
| Sqft\_living | Living area in square feet. |
| Sqft\_lot | Lot size in square feet. |
| Floors | Number of floors. |
| Waterfront | Binary indicator of waterfront presence. |
| View | Quality of the view from the property. |
| Condition | Condition of the house. |
| Grade | Overall grade given to the housing unit. |
| Sqft\_above | Square footage of house apart from basement. |
| Sqft\_basement | Square footage of the basement. |
| Yr\_built | Year the house was built. |
| Yr\_renovated | Year the house was renovated. |
| Zipcode | ZIP code of the location. |
| Lat | Latitude coordinate. |
| Long | Longitude coordinate. |
| Sqft\_living15 | Living area in 2015 (implies recent renovations). |
| Sqft\_lot15 | Lot size in 2015. |

Table 1: Features and Description

The major challenges of this report are as follows:

1. Dealing with vast data:
2. Applying the right business logic for tailored outcomes:
3. Feature Selection & Multicollinearity: Identifying the most important predictors while addressing collinearity issues among highly correlated variables (e.g., sqft\_living, sqft\_above, and sqft\_living15).
4. Generalization & Overfitting – Ensuring that the model generalizes well to unseen data without overfitting to the training dataset.
5. Selecting the best model

The key Objectives of this report are as follows:

1. Model Performance & Evaluation – Determining the best model by comparing Multiple Linear Regression (MLR) and K-Nearest Neighbors (KNN) in terms of prediction accuracy and RMSE (Root Mean Squared Error).
2. Hyperparameter Tuning – Choosing the optimal number of neighbors (K) for KNN classification to improve prediction accuracy.
3. Alternative Algorithm Exploration – Investigating additional algorithms (e.g., Ridge Regression, Lasso, Decision Trees) beyond standard course materials and comparing their performance with MLR and KNN.
4. *Literature Review*

House price prediction is a critical area in real estate analytics, financial modeling, and urban planning. Researchers have utilized both statistical and machine learning approaches to improve accuracy and reliability.

* 1. Traditional Regression Approaches

Multiple Linear Regression (MLR) is one of the earliest methods used for house price prediction, leveraging features such as square footage, number of bedrooms, and location (Li & Brown, 1980). However, its performance is often affected by multicollinearity and the inability to capture non-linear relationships (Malpezzi, 2003).

To address these issues, regularization techniques such as Ridge and Lasso Regression have been introduced. Ridge regression reduces the influence of correlated predictors by applying an L2 penalty, while Lasso regression performs both regularization and feature selection by shrinking some coefficients to zero (Tibshirani, 1996; Zou & Hastie, 2005). These methods improve generalization and model interpretability.

* 1. Machine Learning Models

With advancements in computational power, researchers have explored non-linear machine learning techniques to enhance prediction accuracy (Kok et al., 2017).

1. K-Nearest Neighbors (KNN): A non-parametric model that estimates house prices based on similarity with neighboring properties. While it is effective for small datasets, it struggles with high-dimensional data and requires careful feature scaling (Bishop, 2006; Zhang et al., 2015).
2. Decision Trees & Random Forests: Decision trees segment data using hierarchical rules, capturing non-linear relationships. Random Forest, an ensemble of decision trees, improves accuracy by reducing overfitting through bagging (Breiman, 2001; Qu et al., 2020).
3. Gradient Boosting (GBM) & XGBoost: These boosting algorithms iteratively correct errors from previous models, often outperforming traditional regression methods. However, they require hyperparameter tuning to avoid overfitting (Chen & Guestrin, 2016; Hastie et al., 2009).
   1. Feature Engineering & Categorical Variables

Handling categorical variables such as zip codes and house conditions is crucial for model performance. Techniques like one-hot encoding, target encoding, and ordinal encoding help convert categorical features into numerical values (Cai et al., 2019). Additionally, dimensionality reduction techniques such as Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE) are used to retain only the most relevant predictors while improving computational efficiency (Guyon & Elisseeff, 2003).

* 1. Regression vs. Machine Learning: A Comparative Analysis

Comparative studies highlight trade-offs between interpretability and predictive accuracy. MLR provides a clear understanding of feature importance but struggles with complex interactions (Liang et al., 2020). On the other hand, machine learning models such as Random Forest and XGBoost capture non-linear relationships more effectively, often yielding lower prediction errors. KNN performs well in localized housing markets but is computationally expensive for large datasets (Ahmed & Moustafa, 2021).

* 1. Applications of House Price Prediction

Accurate house price prediction has practical applications in various domains:

1. Real estate agencies & brokers use predictive models to assess property values and guide buyers and sellers.
2. Financial institutions leverage these models for mortgage risk assessment and loan approvals (Agarwal et al., 2019).
3. Urban planners & policymakers use price predictions to track market trends, housing affordability, and infrastructure development.
4. Academic research continues to refine methodologies, ensuring better forecasting models for future real estate markets.
5. *Pre-processing and Exploratory Data Analysis*
   1. An Overview of the CRISP-DM framework*.*

CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, is a cyclical process that provides a systematic approach to planning, organizing, and implementing a data mining activities in a project (Chumbar, 2023).

It has six major phases:

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

Diagram of a diagram of data

AI-generated content may be incorrect.

Fig 1: CRISP-DM Framework

In this report, we have followed the CRISP-DM framework to structure our data mining work and avoid getting lost in the process.

**CRISP-DM: Business and Data Understanding Stage**

In the real estate domain, where price is crucial to all stakeholders involved, we understand that the price depends on various factors, such as the house’s size, location, and condition. However, we need to determine which factors are the most important. Although each feature gives a unique and rather important insight, we cannot retain all of them as we aim to achieve a perfect model. Hence, no model is perfect and there is always room for better accuracy.

**CRISP-DM: Data Preparation (Preprocessing and EDA) Stage**

Although the data was relatively clean and organized, we found a few missing values in features yr\_renovated and sqft\_above. We replaced NaN with 0 assuming these are non-renovated houses and imputed the missing values with mean strategy respectively.

Our mean price is approximately 540,000 USD, which we set as a benchmark for our RMSE scores. Our initial goal was to keep the RMSE within 10% to 20% of this mean price for all our models. Whether we were able to achieve this range will be discussed further in this report.

The date feature had date stored in ISO 8601 basic format which is not easy to work with. Hence, we converted it to DATETIME and extracted day, month and year for our understanding.

Most machine learning algorithms operate optimally with numerical data. Therefore, the categorical features (‘waterfront’, ‘view’, and ‘zipcode’) in our dataset were transformed into a suitable format for model training using a technique called one-hot encoding. This process is crucial for enabling the model to accurately interpret and utilize categorical information during both training and prediction.

From the summary statistics, we picked potential outliers in the bedrooms and bathrooms features. We employed a matplotlib plot to visualize this as follows:

A comparison of a bar graph

AI-generated content may be incorrect.

Fig 2: Outliers Boxplot: Bedrooms and Bathrooms

Properties with 33 bedrooms and zero bedrooms or bathrooms were removed, as these were considered data errors or unusual cases that could negatively influence model performance.

Houses with extremely small living spaces (under 400 sq ft) were filtered out, focusing on typical residential properties and mitigating potential bias from unusual property types.

Very large lot sizes (over 100,000 sq ft) were also excluded, under the assumption that these represent outliers or different land use cases outside the scope of typical residential properties.

The ‘id’ column was removed since it serves as a unique identifier and does not contribute to price prediction.

We ran code for correlation analysis and obtained the following result:

price 1.000000

sqft\_living 0.700814

grade 0.664051

sqft\_above 0.600292

sqft\_living15 0.584798

bathrooms 0.519219

sqft\_basement 0.327700

bedrooms 0.314367

lat 0.312337

view\_4 0.312318

zipcode\_98004 0.275420

waterfront\_1 0.272145

floors 0.252234

zipcode\_98039 0.215522

zipcode\_98040 0.208286

Fig 3: Correlation Analysis (Price)

Primary Influencers:

1. Living Space: Sqft\_living (0.70) is the most dominant factor—larger homes command higher prices.
2. House Grade & Above-Ground Space: Grade (0.66) and sqft\_above (0.60) significantly impact value.
3. Neighborhood & Features: Nearby home size (0.58) and bathrooms (0.52) also play key roles.

Secondary influencers:

1. Basement & Bedrooms: Sqft\_basement (0.33) and bedrooms (0.31) contribute modestly.
2. Location & View: Latitude (0.31), waterfront (0.27), and specific views (0.31) affect prices.
3. Zip Code Effects: Zip codes like 98004 (0.28), 98039 (0.22), and 98040 (0.21) show weak positive correlations with price.

While correlation doesn’t imply causation, these insights help identify key factors for price prediction.

**EDA plots:**

**Price Distribution & Outliers:**

A graph of a house price

AI-generated content may be incorrect.Fig 4: Price Distribution and Outliers

A graph of a house price

AI-generated content may be incorrect.

Fig 5: Price Distribution and Outliers (Right Skewed)

The initial distribution of house prices was highly right-skewed, meaning that a small number of extremely high-priced houses created a long tail in the dataset. This skewness can negatively impact regression models, leading to biased predictions and suboptimal performance.

To address this issue, we applied a natural logarithm (log) transformation to the target variable (price). This transformation compresses large values while preserving the order of the data. The new transformed variable (log\_price) has a more symmetrical and normal-like distribution.

**Benefits of Log Transformation:**

1. Reduces Skewness: The long tail caused by expensive houses is reduced, making the distribution more balanced.
2. Improves Model Performance: Regression models perform better when the dependent variable is normally distributed, leading to more reliable coefficients and better predictive accuracy.
3. Stabilizes Variance: Log transformation helps mitigate heteroscedasticity.

**Price Distribution & Features:**

A graph of a number of different sizes

AI-generated content may be incorrect.

Fig 5: Before and After Log Transformation

A graph of a graph

AI-generated content may be incorrect.

A graph of a graph

AI-generated content may be incorrect.

Fig 6: Price vs Features Before Log Transformation

A graph with blue lines

AI-generated content may be incorrect.

A graph with blue lines

AI-generated content may be incorrect.

Fig 7: Price vs Features After Log Transformation

**Key Observations:**

1. Price vs. Sqft Living: A non-linear trend was observed, with larger homes showing greater price variation, leading to heteroskedasticity.
2. Price vs. Grade: A step-like pattern indicated a categorical influence, with higher grades correlating with higher but more variable prices.
3. Price vs. Bathrooms: More bathrooms generally led to higher prices, though the relationship showed significant spread.

Why Apply Log Transformation?

1. Price was right-skewed, with a few expensive homes dominating the scale.
2. Regression models assume linear relationships, while EDA suggested non-linearity.

Applying log(price) helped:

1. Normalize the distribution for clearer patterns.
2. Reduce heteroskedasticity for a better model fit.
3. Improve interpretability, as coefficients now represent percentage changes.

After the log transformation, relationships appeared more linear, making them more suitable for regression analysis. We dropped the original price column

1. *Feature Selection and Models*

**CRISP-DM: Modelling Stage**

**Phase 1: Multiple Linear Regression Models**

Metrics used for performance evaluation (GeeksforGeeks, 2024):

1. Mean Absolute Error (MAE): A measurement of the typical absolute discrepancies between a dataset's actual values and projected values.

A number and number symbols

AI-generated content may be incorrect.

Fig 8: MAE

1. Mean Squared Error (MSE): It measures the square root of the average discrepancies between a dataset's actual values and projected values.

A black and white math symbol

AI-generated content may be incorrect.

Fig 9: MSE

1. R-squared (R²) Score: It is a useful statistic for evaluating the overall effectiveness and explanatory power of a regression model.

A black number on a white background

AI-generated content may be incorrect.

Fig 10: (R²) Score

1. Root Mean Squared Error (RMSE): It is used to measure the accuracy or goodness of fit of a predictive model, especially when the predictions are continuous numerical values.

A black and white math symbol

AI-generated content may be incorrect.

Fig 11: RMSE

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Model 0** | **Model 1** | **Model 2** |
| MAE | 0.1284 | 0.2796 | 0.2794 |
| MSE | 0.0298 | 0.1200 | 0.1188 |
| (R²) Score | 0.8900 | 0.5570 | 0.5612 |
| RMSE | 0.1726 | 0.3464 | 0.3447 |

Table 2: Performance Metrics (MLR)

**Key Observations:**

* Model 0 significantly outperforms Models 1 and 2 in every metric, indicating that including all features is beneficial for housing price prediction.
* Model 1 vs. Model 2: The addition of sqft\_living15 in Model 2 slightly improves MSE and R² but has minimal impact overall, suggesting its contribution is marginal.
* R² Score Drop: Removing features in Model 1 causes a substantial drop in R² from 0.89 (Model 0) to ~0.56 (Model 1 and 2), meaning nearly half the explanatory power is lost.
* Model 2’s Slight Improvement: The very small gain in R² (+0.0042) and minor reduction in RMSE (~0.0017) indicate that sqft\_living15 is correlated with price (0.6184), but other missing features are likely more critical.

**Implications:**

* The missing features in Models 1 and 2 contain crucial information. The large drop in R² suggests these features significantly contribute to housing price variation.
* sqft\_living15 helps but does not compensate for the removed features. Feature selection should focus on identifying the most valuable variables rather than just adding correlated ones.
* Model 0 should be the preferred model unless there are constraints (e.g., interpretability, computation time).

**Feature Importance and Coefficient Analysis:**

1. Top 6 Features in Model 0

|  |  |
| --- | --- |
| **Feature** | **Coefficient** |
| sqft\_living | 0.3833 |
| zipcode\_98004 | 0.1193 |
| grade | 0.1053 |
| zipcode\_98115 | 0.1042 |
| zipcode\_98103 | 0.1010 |
| zipcode\_98112 | 0.0981 |

Table 3: Best Features

**Interpretation:**

* **sqft\_living (0.3833)** is the **strongest positive predictor**, meaning larger living spaces strongly drive housing prices.
* **Zip codes play a significant role**, with some areas (e.g., 98004, 98115) having a high positive impact on price. This suggests that **location is as important as property features**.
* **grade (0.1053)**, which represents building quality, is another key driver. Homes with better materials and finishes command higher prices.

1. Coefficients for Model 1

|  |  |
| --- | --- |
| **Feature** | **Coefficient** |
| grade | 0.205747 |
| sqft\_living | 0.000320 |
| sqft\_above | -0.000320 |
| Bathrooms | -0.018811 |

Table 4

**Interpretation:**

* **grade (0.2057) is now the dominant feature**, meaning in the absence of location-based variables, housing quality becomes the most important factor.
* **sqft\_living’s coefficient (0.00032) drops dramatically** compared to Model 0 (0.3833). This suggests that sqft\_living’s effect is being captured by other missing variables in Model 0.
* **Negative impact of sqft\_above (-0.00032) and bathrooms (-0.0188)** suggests that, when considered in isolation, these features do not always contribute positively to price.

1. Coefficients for Model 2

|  |  |
| --- | --- |
| **Feature** | **Coefficient** |
| grade | 0.189924 |
| sqft\_living | 0.000291 |
| sqft\_living15 | 0.000090 |
| sqft\_above | -0.000146 |
| bathrooms | -0.012480 |

Table 5

**Interpretation:**

* **grade remains the strongest feature (0.1899)** but its importance slightly decreases compared to Model 1 (0.2057).
* **sqft\_living15 (0.000090) has a very small coefficient**, meaning its effect on price is weak after accounting for other variables.
* **Negative coefficients for sqft\_above and bathrooms remain**, reinforcing that their impact may depend on other interactions.

log\_price 1.000000

grade 0.700259

sqft\_living 0.693751

sqft\_living15 0.618408

sqft\_above 0.596018

bathrooms 0.545301

Fig 12: Correlation Analysis (log\_price)

**Coefficient Analysis with Correlation Insights:**

**Strong correlations with price**:

* grade (**0.7002**) and sqft\_living (**0.6938**) are the **two most influential factors** in determining price.
* sqft\_living15 (**0.6184**) is also strongly correlated but weaker than sqft\_living.

**sqft\_living vs. sqft\_living15**:

* Although sqft\_living15 has a strong correlation, **its coefficient in Model 2 is small**, suggesting that its influence is **already captured by sqft\_living**.

**sqft\_above and bathrooms have lower correlations (~0.55-0.59)**:

* This indicates they contribute to price but are **not as critical as grade or living space**.

**Key Takeaways:**

* **In Model 0, sqft\_living is a dominant predictor, but its effect weakens in Models 1 and 2**, suggesting that its influence is tied to other missing features.
* **Location (zip codes) is crucial in Model 0**, but its removal in Models 1 and 2 significantly reduces predictive power.
* **Adding sqft\_living15 has only a minor effect**, meaning it is somewhat useful but **not a substitute for the missing features**.

**Phase 2: Regularization Techniques**

Ridge and Lasso Regression are widely used machine learning techniques for regularizing linear models to prevent overfitting and enhance predictive accuracy. Both methods incorporate a penalty term into the model’s cost function to constrain the coefficients, but they differ in how this penalty is applied.

Ridge Regression, also called L2 regularization, imposes a penalty on the squared values of the coefficients. In contrast, Lasso Regression, known as L1 regularization, applies a penalty based on the absolute values of the coefficients (GeeksforGeeks, 2025).

|  |  |  |  |
| --- | --- | --- | --- |
| **Metric** | **Model 0 (Best)** | **Ridge** | **Lasso** |
| RMSE | 0.1726 | 0.3423 | 0.3557 |

Table 6

**Analysis:**

#### 1. Ridge Regression (RMSE: 0.3423):

* RMSE increased compared to MLR (0.1726 → 0.3423), indicating that Ridge did not improve the model's predictive performance in this case.
* This suggests that the original MLR model was already performing well, and the penalty term in Ridge may have over-regularized the coefficients, reducing their effectiveness.
* While Ridge helps in handling multicollinearity, it seems to have come at the cost of higher prediction errors.

#### 2. Lasso Regression (RMSE: 0.3557):

* RMSE increased even further compared to Ridge (0.3423 → 0.3557), showing that Lasso’s feature selection may have removed some important predictors.
* Since Lasso shrinks some coefficients to zero, it’s likely that certain features that were contributing to lower RMSE in MLR were eliminated.
* This suggests that Lasso might not be the best choice for this dataset, as it has led to higher prediction errors.

#### **Interpretation:**

* MLR (0.1726 RMSE) remains the best model in terms of predictive accuracy.
* Ridge (0.3423 RMSE) increased RMSE significantly, indicating possible over-regularization.
* Lasso (0.3557 RMSE) performed the worst, suggesting that the removal of some features hurt predictive accuracy.
* Regularization did not improve performance here, likely because the MLR model was already well-tuned and applying penalties reduced the effectiveness of important features.

### **Key Takeaways:**

* Regularization is usually helpful when dealing with overfitting, but in this case, it seems that MLR was already performing optimally.
* The high RMSE for Ridge and Lasso indicates that the penalty terms negatively affected model performance.
* Lasso’s feature selection may have removed important variables, leading to a further increase in RMSE.
* For this dataset, sticking with MLR might be the best option, unless hyperparameter tuning is done to optimize Ridge and Lasso.

**Phase 3: KNN Model**

The evaluation metrics for KNN are the same as those used in MLR, ensuring consistency in model comparison. These include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared (R²) Score

**Selection of K Values:**

Based on the project requirements, we evaluated **KNN models with K = 5 and K = 10** to determine which provides better prediction accuracy.

|  |  |  |
| --- | --- | --- |
| **Metric** | **K=5** | **K=10** |
| MAE | 0.1492 | 0.1480 |
| MSE | 0.0420 | 0.0416 |
| (R²) Score | 0.8447 | 0.8462 |
| RMSE | 0.2051 | 0.2041 |

Table 7: Performance Metrics (KNN)

**Analysis of Metrics:**

* K = 10 outperforms K = 5 in all metrics, achieving a slightly lower RMSE (0.2041 vs. 0.2051) and a higher R² score (0.8462 vs. 0.8447).
* The differences are minor, indicating that both values of K yield similar predictive performance.
* Increasing K results in smoother predictions, reducing noise but potentially losing detail about local trends.
* K = 10 is the preferred model, as it provides the most accurate predictions with slightly lower errors.

**Feature Importance in KNN:**

Unlike tree-based models like Random Forest, KNN does not assign feature importance scores because it does not learn explicit relationships between features and the target variable. Instead, it relies purely on distance calculations in the feature space.

**Interpretation**:

* KNN is a useful baseline model for house price prediction, particularly in scenarios where local property trends are essential.
* K = 10 performed slightly better than K = 5, making it the optimal choice for predictions in this dataset.
* However, KNN is computationally expensive for large datasets, as it requires distance calculations for every new prediction.
* Further tuning of K and feature selection could enhance performance, but other models like Random Forest or Gradient Boosting may be more suitable for larger datasets.

**Phase 4: Random Forest Model**

The same metrics used in MLR and KNN are applied here for consistent comparison:

* Mean Absolute Error (MAE)
* Mean Squared Error (MSE)
* Root Mean Squared Error (RMSE)
* R² Score

|  |  |
| --- | --- |
| **Metric** | **Random Forest Model** |
| MAE | 0. 1199 |
| MSE | 0. 0281 |
| (R²) Score | 0. 8963 |
| RMSE | 0.1676 |

Table 8: Performance Metrics (Random Forest Model)

**Analysis of Metrics:**

* Best-performing model so far with the lowest RMSE (0.1676) and highest R² (0.8963).
* Significant improvement over KNN and MLR, demonstrating the strength of ensemble learning.
* Outperforms KNN because it captures complex feature interactions rather than relying solely on distance-based similarity.
* Better generalization than linear models, as it does not rely on assumptions about data distribution.

**Feature Importance:**

* Unlike models like MLR and KNN, Random Forest (RF) provides feature importance scores, which help in understanding which factors influence house prices the most.
* Feature importance is calculated based on how much each variable reduces prediction errors when used in decision trees.

|  |  |
| --- | --- |
| **Feature** | **Coefficient** |
| sqft\_grade | 0.708213 |
| Sqft\_bathrooms | 0.124580 |
| grade | 0.055653 |
| Sqft\_living\_squared | 0.039866 |
| Sqft\_living | 0.039796 |
| zipcode\_98004 | 0.025457 |
| zipcode\_98039 | 0.006435 |

Table 8: Important Features

**Insights from Feature Importance Analysis**

* sqft\_grade (0.7082) is the most influential factor, meaning house prices are heavily impacted by both size (sqft\_living) and quality (grade).
* sqft\_bathrooms (0.1246) also plays a significant role, reinforcing the idea that more bathrooms increase property value.
* zipcode\_98004 and zipcode\_98039 (premium locations) are important but contribute less than structural attributes like size and quality.
* sqft\_living\_squared (0.0399) suggests a non-linear effect of house size, meaning that the price increase slows down as houses get larger.

**Interpretation:**

* Random Forest is currently the best model due to its superior predictive power and ability to capture complex interactions.
* Its high R² score (0.8963) shows that it explains most of the variance in house prices.
* **Downside:** It is computationally expensive, making it less suitable for real-time applications compared to simpler models.
* **Future improvements:** Hyperparameter tuning (adjusting the number of trees, depth, and features per split) could further enhance performance.

**Phase 5: SVM Model**

The same metrics used in previous models are applied here:

* Mean Absolute Error (MAE)
* Mean Squared Error (MSE)
* Root Mean Squared Error (RMSE)
* R² Score

|  |  |
| --- | --- |
| **Metric** | **Random Forest Model** |
| MAE | 0. 1275 |
| MSE | 0. 0297 |
| (R²) Score | 0. 8902 |
| RMSE | 0. 1725 |

Table 9: Performance Metrics (SVM)

**Analysis of Metrics:**

* Performs worse than Random Forest (higher RMSE and lower R² score), but still better than KNN and MLR.
* Slightly higher RMSE than Random Forest (0.1725 vs. 0.1676), indicating marginally lower accuracy.
* Better generalization than KNN, as it does not rely on direct instance comparisons.
* More computationally expensive than MLR and KNN due to the optimization process.

**Feature Importance:**  
Unlike tree-based models, SVM does not provide feature importance scores. However, certain kernel tricks (e.g., polynomial or RBF kernels) can be used to capture non-linear interactions between house features (Hastie et al., 2009).

**Interpretation:**

* SVM is a competitive model, offering good accuracy but slightly lagging behind Random Forest.
* Random Forest remains the best choice for this dataset due to its lower RMSE and higher R² score.
* Further improvements: Hyperparameter tuning (e.g., adjusting C, epsilon, and kernel type) could enhance SVM performance.

1. *Evaluation of Approaches*

**CRISP-DM: Evaluation Stage**

To determine the best predictive model for house price estimation, we evaluated Multiple Linear Regression (MLR), K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM). The evaluation was based on Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), and R² Score.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **RMSE** | **MAE** | **MSE** | **R² Score** | **Key Observation** |
| **MLR (Best Variant)** | 0.1726 | 0.2694 | 0.0298 | 0.5930 | Baseline linear model struggles with complex data |
| **KNN (K=10)** | 0.2041 | 0.1480 | 0.0416 | 0.8462 | Distance-based, computationally expensive |
| **Random Forest** | 0.1676 | 0.1199 | 0.0281 | 0.8963 | Best model, handles non-linearity, strong generalization |
| **SVM** | 0.1725 | 0.1275 | 0.0297 | 0.8902 | Competitive but slightly worse than Random Forest |

Table 10: Model Performance Comparison

**Comparison & Analysis:**

a) Multiple Linear Regression (MLR):

* RMSE (0.1726) is higher than RF but lower than KNN, indicating moderate performance.
* Struggles with non-linearity in house price prediction.
* Sensitive to multicollinearity, as seen with correlated features (e.g., sqft\_living and sqft\_above).
* Good baseline model but limited in capturing complex interactions.

b) K-Nearest Neighbors (KNN):

* Higher RMSE (0.2041) and MSE than other models, showing weaker predictive power.
* Sensitive to data scaling, making it computationally expensive.
* K=10 performed better than K=5, indicating that larger neighborhoods improve generalization.
* Not the best choice for high-dimensional datasets, as it becomes inefficient.

c) Random Forest (RF):

* Best-performing model with the lowest RMSE (0.1676) and highest R² score (0.8963).
* Handles non-linearity well, making it highly effective for real estate predictions.
* Feature importance analysis shows that variables like sqft\_living and grade contribute significantly.
* Computationally intensive but justifies the performance improvement.

d) Support Vector Machine (SVM):

* RMSE (0.1725) is slightly higher than Random Forest, but better than MLR and KNN.
* Handles complex relationships but is computationally expensive.
* Feature importance is not directly available, unlike RF.
* It slightly underperforms compared to RF, suggesting the need for hyperparameter tuning.

**Interpretation**:

* Random Forest is the best model for this dataset, with the lowest RMSE and highest R² score.
* SVM is a strong alternative, though slightly less accurate than RF.
* KNN is computationally expensive and does not perform well on high-dimensional data.
* MLR serves as a useful baseline but is outperformed by non-linear models.
* Future work: Hyperparameter tuning (e.g., optimizing RF depth, adjusting SVM kernel parameters) could further refine results.

1. *Additional Considerations*

**Feature Selection Using Recursive Feature Elimination (RFE)**

Why does Feature selection matter?

* Machine learning models can suffer from redundant or irrelevant features, leading to overfitting and increased computational cost.
* Recursive Feature Elimination (RFE) helps by iteratively removing the least important features to retain only the most relevant ones.

**Applying RFE to Multiple Linear Regression (MLR)**

Steps Taken:

* RFE was applied to MLR to reduce multicollinearity and improve interpretability.
* The model was trained using all features, and the least significant ones were removed in multiple iterations.
* Final Selected Features: sqft\_living, grade, sqft\_living\_squared, sqft\_grade, sqft\_bathrooms, zipcode\_98004, zipcode\_98039

**Impact of RFE on Model Performance**

|  |  |  |
| --- | --- | --- |
| **Model** | **RMSE** | **Key Observation** |
| MLR (All Features) | 0.1726 | More accurate but includes redundant features |
| MLR with RFE | 0.3423 | More interpretable, but higher RMSE |

Table 11: Comparison of MLR Before and After RFE

**Key Insights:**

* RFE simplified the model, making it more interpretable.
* However, removing too many features led to increased RMSE, suggesting that some excluded features were still valuable.
* For MLR, feature selection should be balanced to maintain both accuracy and interpretability.

1. *Conclusion*

This research focused on creating machine learning models for estimating house prices with MLR, KNN, Random Forest, and SVM models. The Random Forest model was determined to be optimal for this dataset because it had the lowest RMSE of 0.1676 and the highest R² score of 0.8963.

**Key Findings:**

* MLR is a useful baseline approach; however, its performance was limited due to multicollinearity and failing to account for non-linear relationships.
* KNN was less accurate because it did not work well in high dimensions, resulting in high RMSE and poor computational efficiency.
* Regularized MLR models using Ridge or Lasso did not outperform the baseline models.
* SVM gave good accuracy but was the least accurate and most expensive compared to the other models.
* Feature selection plays a crucial role, but excessive reduction (as seen with RFE in MLR) can lead to performance trade-offs.

**Final Recommendation:**

When considering practical purposes of house price prediction, the Random Forest model is most effective due to its high accuracy, ability to capture complex interactions between variables, and robustness to overfitting. The model can be further improved by tuning hyperparameters, optimizing features, and adding external economic variables (interest rates, inflation) to improve accuracy.

1. *References*
2. Li, W., & Brown, H. J. (1980). The impact of housing characteristics on house prices. Journal of Urban Economics, 7(2), 159-176.
3. Malpezzi, S. (2003). Hedonic pricing models: A selective and applied review. Housing Economics and Public Policy, 67-89.
4. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.
5. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301-320.
6. Kok, S., Kopczewska, K., & Mulley, C. (2017). Machine learning methods in house price prediction. Computers, Environment and Urban Systems, 65, 49-61.
7. Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.
8. Zhang, Z., Jin, R., & Zhou, Z. H. (2015). Understanding k-nearest neighbor performance in high-dimensional spaces. Neural Information Processing Systems (NIPS), 28, 1-9.
9. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
10. Qu, X., Liu, J., & Zhang, H. (2020). Improving house price prediction using ensemble learning techniques. Expert Systems with Applications, 141, 112967.
11. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794.
12. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction. Springer.
13. Cai, W., Ren, F., & Yuan, X. (2019). Impact of categorical variable encoding methods in predictive modeling: An empirical study. Journal of Data Science and Analytics, 11(3), 215-228.
14. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157-1182.
15. Liang, Y., Wang, X., & Zhang, L. (2020). Comparing regression and machine learning models for housing price prediction. Journal of Real Estate Finance and Economics, 60(1), 23-45.
16. Ahmed, M., & Moustafa, A. (2021). Comparative analysis of regression, KNN, and ensemble learning for housing price prediction. Expert Systems with Applications, 163, 113722.
17. Agarwal, R., Ghosh, S., & Sarkar, P. (2019). Predictive modeling in real estate: A mortgage risk assessment approach. Financial Innovation, 5(1), 1-15.
18. Chumbar, S. (2023). The CRISP-DM Process: A Comprehensive Guide. Medium. Retrieved from https://medium.com/@shawn.chumbar/the-crisp-dm-process-a-comprehensive-guide-4d893aecb151.
19. GeeksforGeeks. (2024). Regression Metrics. Retrieved from <https://www.geeksforgeeks.org/regression-metrics/>.
20. GeeksforGeeks. (2025). Ridge Regression vs Lasso Regression. Retrieved from https://www.geeksforgeeks.org/ridge-regression-vs-lasso-regression/.