Document 📓 Presentation 📽️ Linear Regression Demo 📈 Random Forest Regression Demo 🌳 Support Vector Regression Demo 💻

Machine Learning Midterm Report

Introduction/Background

This project aims to use ML techniques to accurately predict the sales price of houses based on various features of the houses.

Literature Review

Several studies have explored ML techniques for house price prediction. One compared Linear Regression, Decision Trees, K-Means, and Random Forest, cor make informed investment decisions [1]. Another study utilized SVMs, Linear Regression, and Random Forest, identifying features like living conditions, traffi advanced approach, deep learning was shown to outperform linear regression but underperform compared to gradient-boosted trees, with the best accurac that applied Random Forest, XGBoost, LightGBM, Hybrid Regression, and Stacked Generalization found that Stacked Generalization offered the most accurat RIPPER, Naïve Bayesian, and AdaBoost identified RIPPER as the best-performing algorithm for predicting whether a house's closing price would exceed its li

Dataset Description

The dataset contains various attributes related to house characteristics in the Ames, Iowa area, with the target variable being `SalePrice`.

Key Features:

- `MSSubClass` : Building class
- `OverallQual` : Overall material and finish quality
- `GrLivArea` : Above-ground living area square feet
- `GarageCars` : Size of garage in car capacity
- `YearBuilt` : Year of original construction
- `SalePrice` : Target variable, the property's sale price

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Conc
0	1	60	RL	65	8,450	Pave	None	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norn
1	2	20	RL	80	9,600	Pave	None	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feed
2	3	60	RL	68	11,250	Pave	None	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norn
3	4	70	RL	60	9,550	Pave	None	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norn
4	5	60	RL	84	14,260	Pave	None	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norn
5	6	50	RL	85	14,115	Pave	None	IR1	Lvl	AllPub	Inside	Gtl	Mitchel	Norn
6	7	20	RL	75	10,084	Pave	None	Reg	Lvl	AllPub	Inside	Gtl	Somerst	Norn
7	8	60	RL	None	10,382	Pave	None	IR1	Lvl	AllPub	Corner	Gtl	NWAmes	PosN
8	9	50	RM	51	6,120	Pave	None	Reg	Lvl	AllPub	Inside	Gtl	OldTown	Arter
9	10	190	RL	50	7,420	Pave	None	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	Arter

Problem Definition

Problem

Predicting the sale price of houses based on their features is a complex task to perform by hand due to the various factors involved, including location, cond

Motivation

Accurate house price predictions can have a significant financial impact. For real estate professionals and potential buyers, understanding what drives hous to faster, data-driven decisions in the housing market.

Methods

Data Preprocessing Methods

1. **Data Cleaning**: We addressed missing values for both numerical and categorical features. For numerical features, missing values were imputed using th central tendency of the data was preserved while avoiding the loss of valuable rows. For categorical features, we replaced missing values with a placeho observations. This approach retained all data points, enabling the model to learn patterns even in cases where categorical information was incomplete.
2. **Feature Engeineering**: We created a new feature called TotalSqFt, which is the sum of 1stFlrSf, 2ndFlrSf, and BsmtSF. We chose to do this because creati house, which is often highly correlated with its value. This engineered feature reduces complexity by combining related square footage features and can home size.
3. **Dimensionality Reduction**: We used dimensionality reduction in certain configurations to reduce the number of features. We specifically used Principal predictive power. This step helped reduce multicollinearity, improve model performance, and make the model more interpretable by focusing only on t allowing for faster model training and testing.

Machine Learning Algorithms

1. **Linear Regression:** We selected Linear Regression for its simplicity, interpretability, and effectiveness in predicting continuous target variables such as h target, allowing us to easily understand how each feature impacts the final price through its coefficients. Linear Regression also works well when multic reduction. Additionally, its low computational cost made it an efficient choice for training on our dataset.
2. **Random Foreset Regression:** We chose to use Random Forest Regression because of its ability to handle complex relationships in the data that might n asusmption about the nature of the relationship between the features and the target would help improve the model's performance. Additionally, Rand predictions from multiple decision trees. This also ensures that it performs well on both training and unseen data.
3. **Support Vector Regression (SVR):** We selected Support Vector Regression because it is a powerful algorithm that can handle non-linear relationships be outliers, which is a useful benefit given our dataset does include several outliers towards the expensive side of the market that the previous models stru hyperparameters and switch between different kernels, which is useful for finding the best fit for our data.

Results and Discussion

Linear Regression Model

Configuration 1

High-Impact Features Without PCA

Numerical Features: `OverallQual`, `GrLivArea`, `TotalBsmtSF`, `YearBuilt`

Categorical Features: `Neighborhood`, `MSZoning`, `BldgType`

PCA: No

R^2 Score

0.83

Root Mean Squared Error (RMSE)

36,152.14

Configuration 1: Model Performance

Analysis: This model performs well due to its inclusion of high-impact features, achieving an R^2 score of 0.83. However, its RMSE suggests further room fo

Configuration 2

Diverse Property Characteristics Without PCA

Numerical Features: `LotFrontage` , `LotArea` , `YearRemodAdd` , `TotalBsmtSF` , `2ndFlrSF` , `BsmtFullBath` , `FullBath`

Categorical Features: `Condition1` , `Condition2` , `RoofStyle` , `HouseStyle` , `MSZoning`

PCA: No

R^2 Score

0.72

Root Mean Squared Error (RMSE)

28,723.24

Configuration 2: Model Performance

Analysis: While this configuration includes a broader range of property characteristics, its predictive power is lower compared to Configuration 1. The low
less relevant for this task.

Configuration 3

Structural and Size-Related Features With PCA

Numerical Features: `OverallQual` , `GrLivArea` , `TotalBsmtSF` , `1stFlrSF` , `GarageCars` , `GarageArea` , `YearBuilt`

Categorical Features: `HouseStyle` , `BldgType` , `RoofMatl`

PCA: Yes, 25 components

R^2 Score

0.68

Root Mean Squared Error (RMSE)

49,708.28

Configuration 3: Model Performance

Analysis: PCA reduces dimensionality, but the performance drops, with an R^2 score of 0.68. This may indicate that some important variance in the data wa

Configuration 4

Balanced Features Without PCA

Numerical Features: `OverallQual` , `YearBuilt` , `TotalBsmtSF` , `GarageArea`

Categorical Features: `Street` , `Neighborhood` , `ExterQual` , `KitchenQual`

PCA: No

R^2 Score

0.80

Root Mean Squared Error (RMSE)

38,926.00

Configuration 4: Model Performance

Analysis: This configuration balances structural and neighborhood-related features, leading to solid performance with an R^2 score of 0.80. Its performanc simpler models.

Configuration 5

Balanced Features With PCA

Numerical Features: `OverallQual` , `YearBuilt` , `TotalBsmtSF` , `GarageArea`

Categorical Features: `Street` , `Neighborhood` , `ExterQual` , `KitchenQual`

PCA: Yes, 35 components

R^2 Score

0.85

Root Mean Squared Error (RMSE)

34,470.69

Configuration 5: Model Performance

Analysis: By applying PCA on a balanced set of features, the model achieves the second-best R^2 score of 0.85. This suggests that dimensionality reduction

Configuration 6

Comprehensive Feature Set Without PCA

Numerical Features: `LotArea` , `OverallQual` , `OverallCond` , `YearBuilt` , `TotalBsmtSF` , `GrLivArea` , `BedroomAbvGr` , `KitchenAbvGr` , `GarageCars` , `GarageAr`

Categorical Features: `Neighborhood` , `BldgType` , `HouseStyle` , `ExterQual` , `GarageType` , `Fence` , `KitchenQual`

PCA: No

R^2 Score

0.87

Root Mean Squared Error (RMSE)

31,947.33

Configuration 6: Model Performance

Analysis: This configuration uses a large feature set without PCA and achieves the best performance, with an R^2 score of 0.87. This indicates that a diverse

Configuration 7

Extended Feature Set Without PCA, With Feature Engineering

Numerical Features: `LotArea` , `OverallQual` , `OverallCond` , `YearBuilt` . `MasVnrArea` , `1stFlrSF` , `GrLivArea` , `BsmtFullBath` , `BsmtHalfBath` , `FullBath` , `E`
`MoSold` , `YrSold` , `TotalSqFt` (feature engineered)

Categorical Features: `Neighborhood` , `BldgType` , `HouseStyle` , `ExterQual` , `LotShape` , `Condition1` , `Condition2` , `SaleType`

PCA: No

R^2 Score

0.88

Root Mean Squared Error (RMSE)

30,220.64

Configuration 7: Model Performance

Analysis: Adding the feature-engineered 'TotalSqFt' improves the performance marginally over Configuration 6. This shows the value of engineering new

The results of our linear regression models across seven configurations highlight the critical impact of feature selection and dimensionality reduction on mo quantitative metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 score.

From the visualizations, we observed that models using a comprehensive and balanced feature set (Configurations 6 and 7) consistently achieved higher R^2 engineered feature, `TotalSqFt`, had the highest R^2 score. This demonstrates that including diverse, relevant features, as well as intelligently engineered feat that models using PCA showed mixed results. While PCA helped reduce dimensionality, it sometimes led to the loss of important variance, as seen in Configu counterparts. However, when paired with a balanced feature set (Configuration 5), PCA showed potential by improving metrics slightly compared to simpler

Overall, for the linear regression machine learning method, we achieved our goals set in the proposal of an R^2 over .7, an MAE of less than $30,000, and a RM

Random Forest Regression Model

Configuration 1

Basic Feature Set Without PCA

Numerical Features: `LotArea`, `OverallQual`, `OverallCond`, `YearBuilt`, `MasVnrArea`, `BsmtFinSF2`, `GrLivArea`, `BsmtFullBath`, `TotRmsAbvGrd`, `WoodDeckS`

Categorical Features: `Neighborhood`, `BldgType`, `HouseStyle`, `ExterQual`, `LotShape`, `Condition1`, `Condition2`, `Electrical`, `SaleCondition`, `PavedDriv`

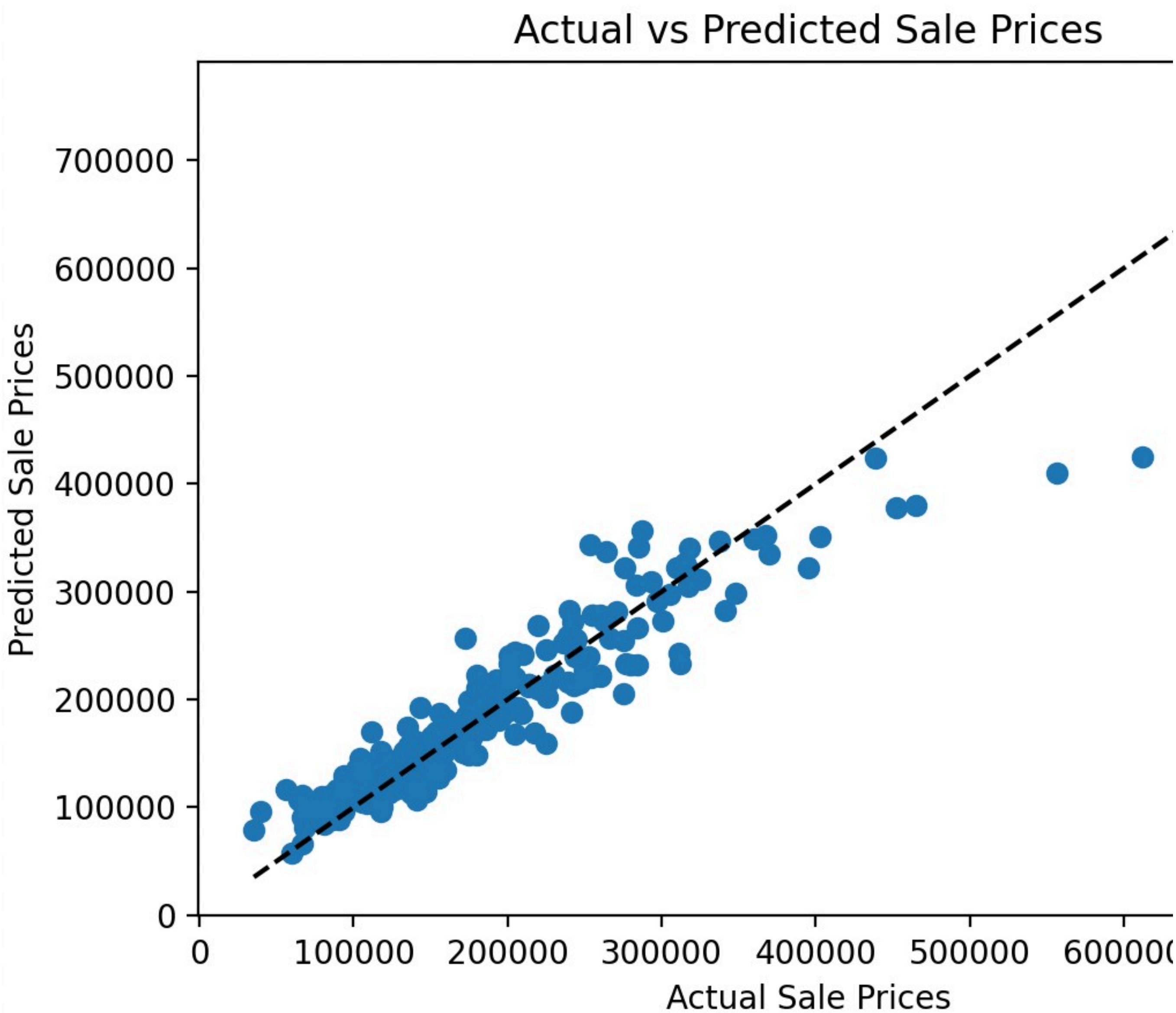
PCA: No

R^2 Score

0.88

RMSE

30,388.60



Configuration 2

Refined Feature Set Without PCA (Best Initial)

Numerical Features: `LotArea` , `OverallQual` , `OverallCond` , `YearBuilt` , `MasVnrArea` , `BsmtFinSF2` , `GrLivArea` , `BsmtFullBath` , `TotRmsAbvGrd` , `WoodDeckS`

Categorical Features: `Neighborhood` , `BldgType` , `HouseStyle` , `ExterQual` , `LotShape` , `Condition1` , `Condition2` , `Electrical`

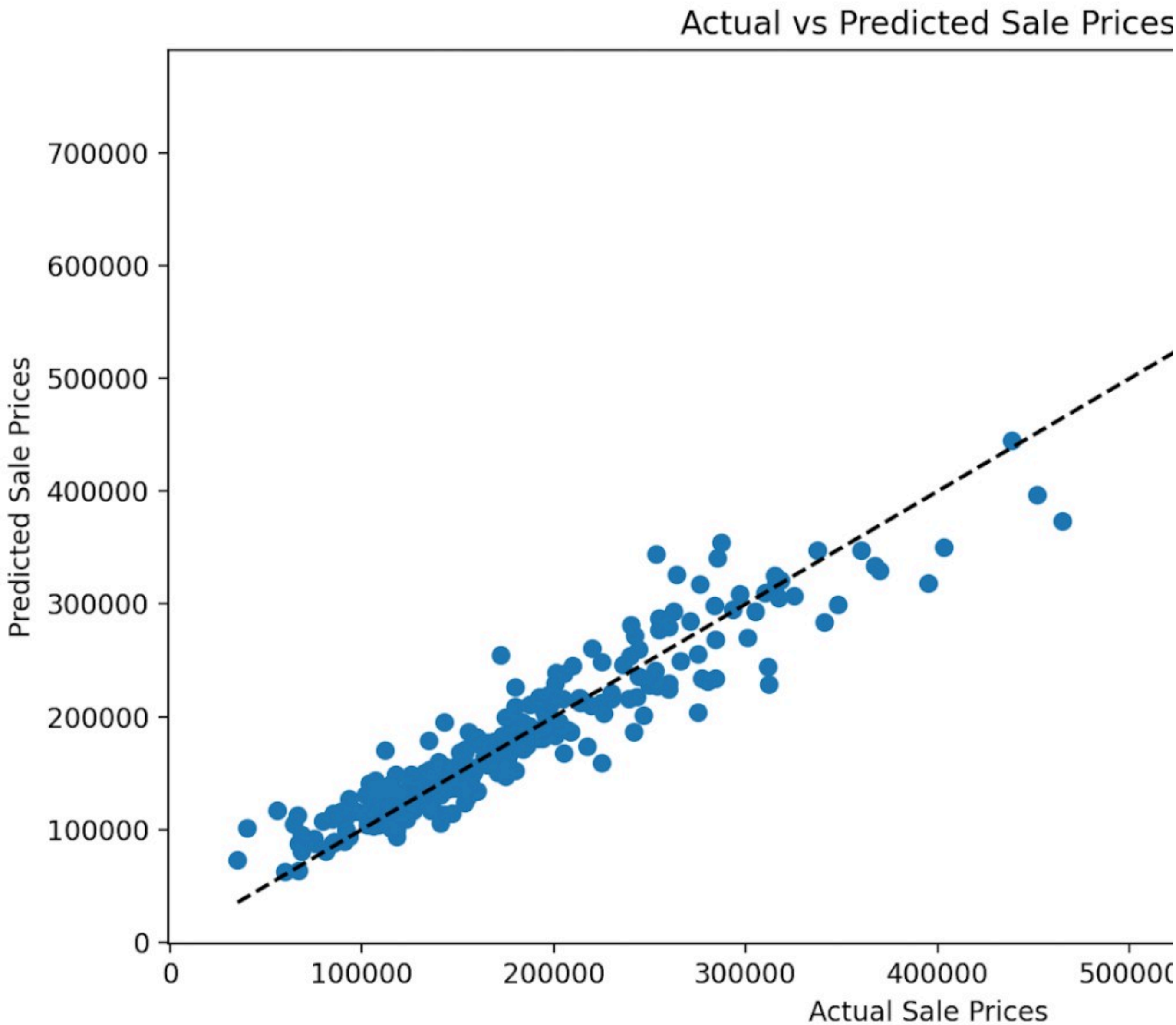
PCA: No

R^2 Score

0.89

RMSE

29,415.73



Configuration 3

Feature Set with PCA

Numerical Features: `LotArea` , `OverallQual` , `OverallCond` , `GrLivArea` , `BsmtFullBath` , `TotRmsAbvGrd` , `WoodDeckSF` , `YrSold` , `ScreenPorch` , `MoSold`

Categorical Features: `Neighborhood` , `BldgType` , `HouseStyle` , `ExterQual` , `LotShape` , `Condition1` , `Condition2` , `Electrical` , `SaleCondition` , `PavedDriv`

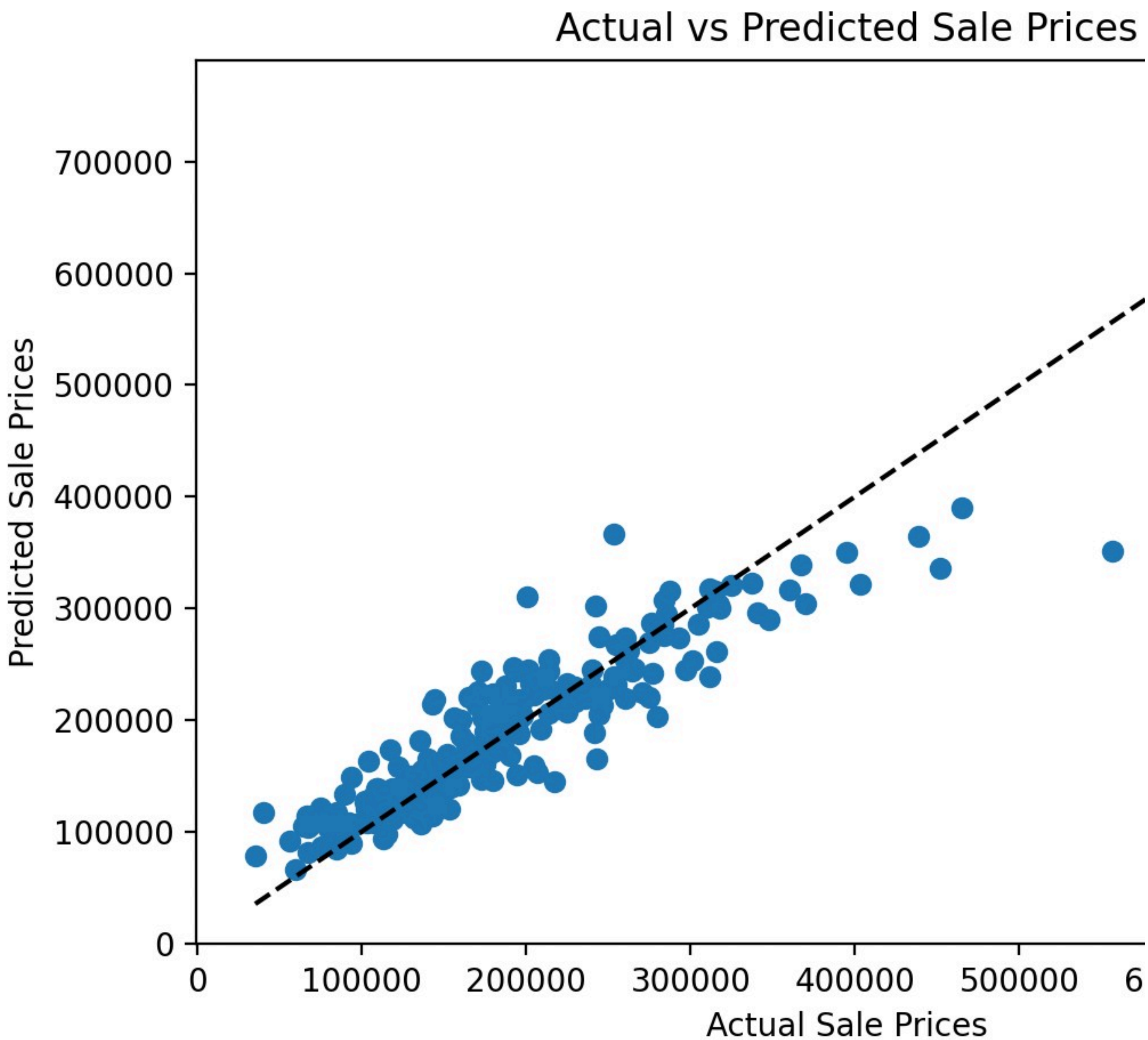
PCA: Yes (87 components)

R^2 Score

0.84

RMSE

34,564.27



Configuration 4

Reduced Feature Set with PCA

Numerical Features: `OverallQual`, `GrLivArea`, `TotalBsmtSF`, `YearBuilt`, `GarageCars`, `GarageArea`, `LotArea`, `TotRmsAbvGrd`, `TotalSqFt`

Categorical Features: `Neighborhood`, `ExterQual`, `KitchenQual`, `BldgType`, `HouseStyle`, `SaleCondition`

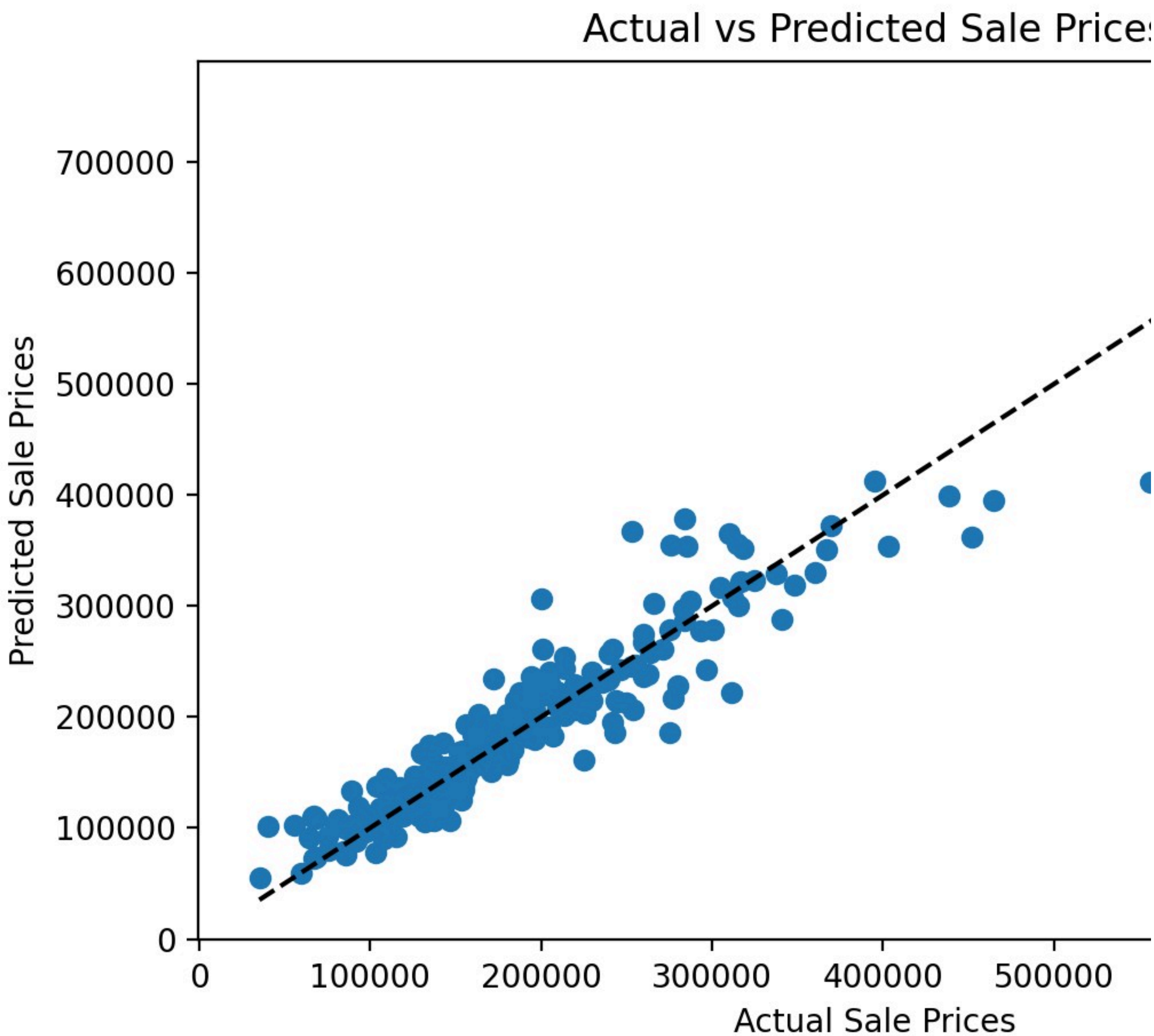
PCA: Yes (10 components)

R^2 Score

0.88

RMSE

29,840.39



Configuration 5

Enhanced Feature Set Without PCA

Numerical Features: `OverallQual` , `GrLivArea` , `GarageCars` , `TotalBsmtSF` , `YearBuilt` , `LotArea` , `1stFlrSF` , `TotalSqFt` , `Qual_GrLiv`

Categorical Features: `Neighborhood` , `ExterQual` , `KitchenQual` , `SaleCondition` , `BldgType` , `HouseStyle`

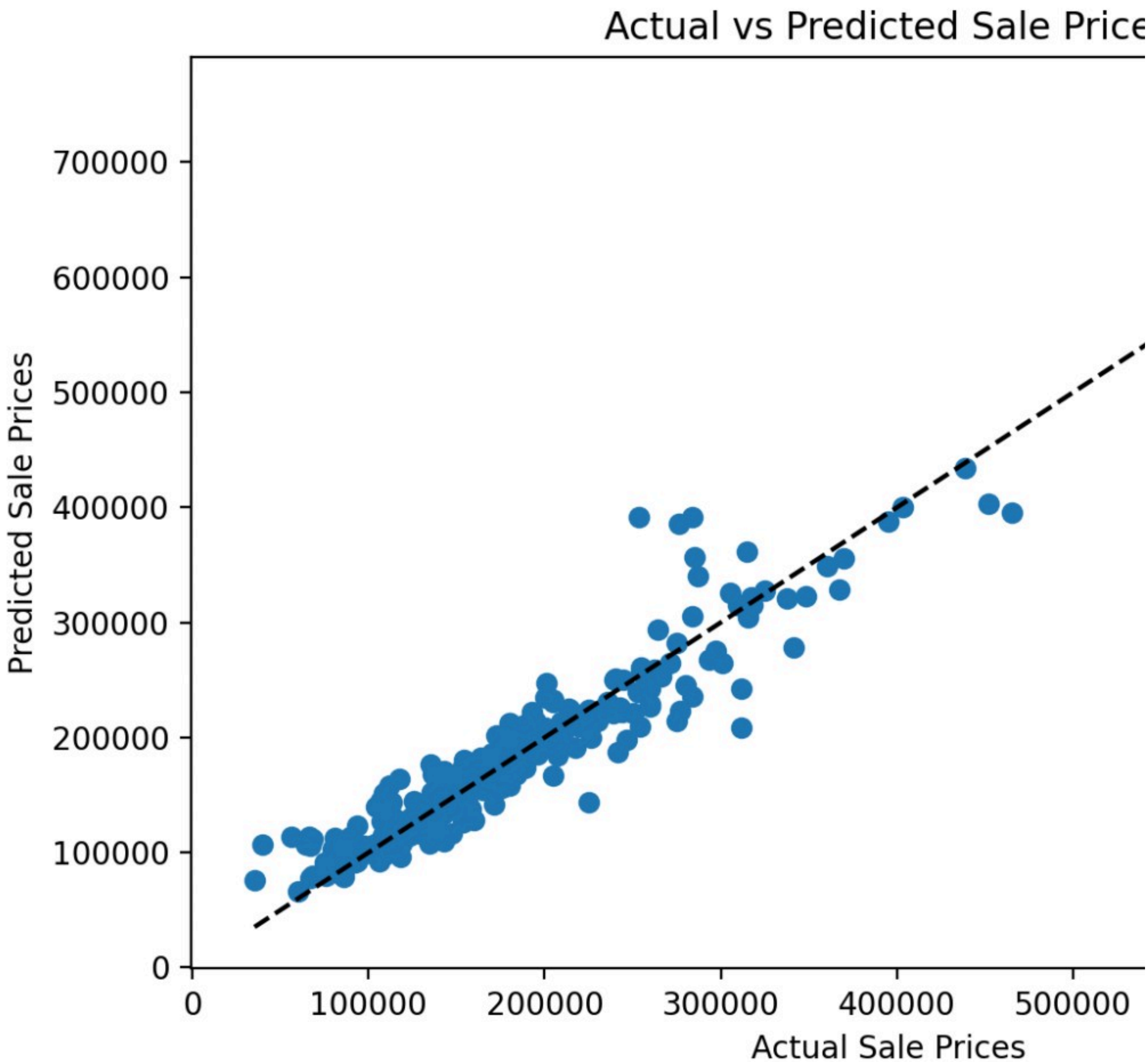
PCA: No

R^2 Score

0.89

RMSE

28,495.45



Configuration 6

Optimized Feature Set (Best Overall)

Numerical Features: `OverallQual` , `GrLivArea` , `TotalBsmtSF` , `GarageCars` , `YearBuilt` , `LotArea` , `MiscVal` , `OverallCond` , `MoSold`

Categorical Features: `CentralAir` , `Neighborhood` , `ExterQual` , `SaleCondition` , `BldgType` , `Foundation` , `BsmtQual` , `ExterCond` , `LandSlope`

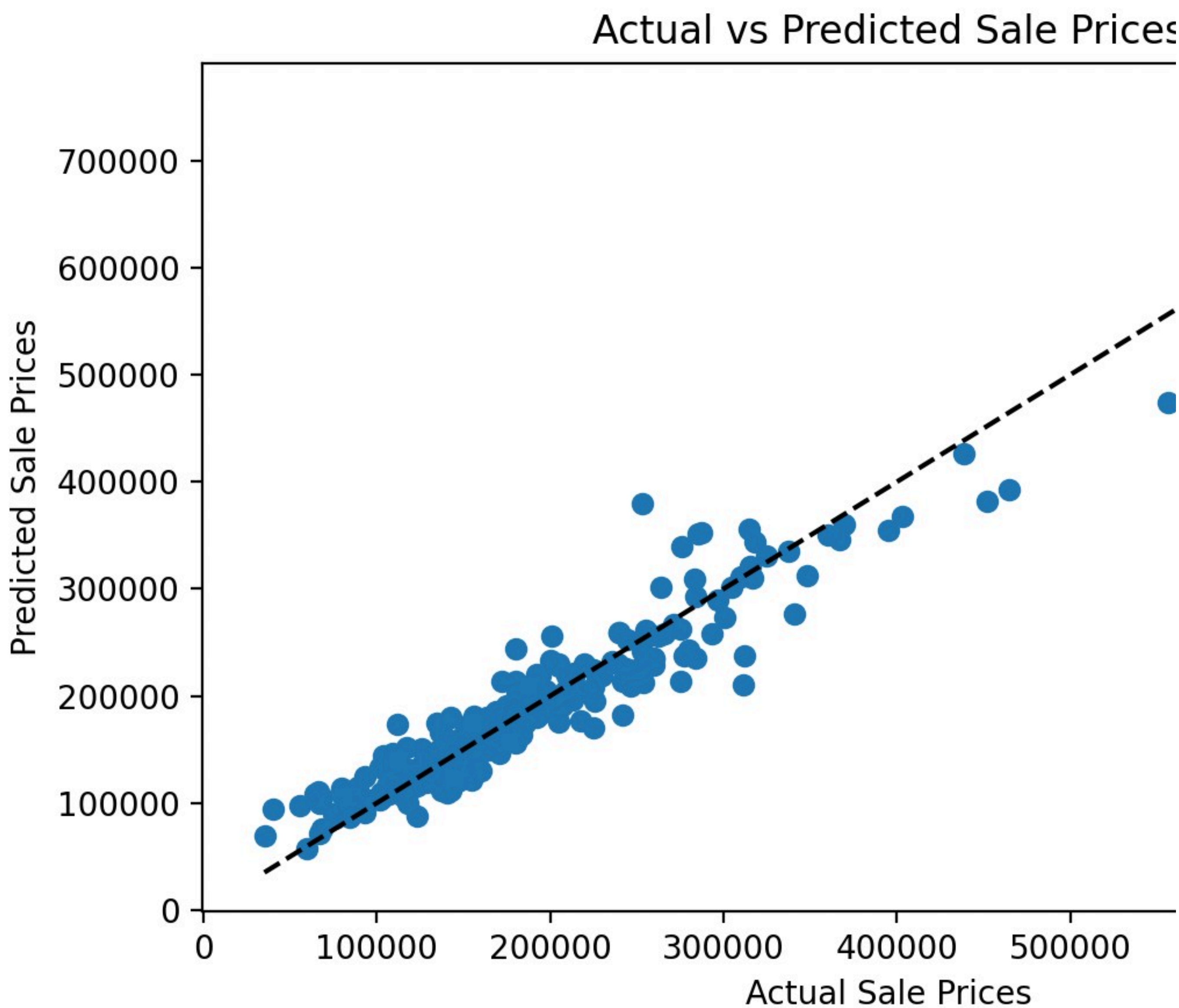
PCA: No

R^2 Score

0.90

RMSE

27,722.53



We tested the Random Forest Regression model across multiple configurations, focusing on feature selection, and the impact of hyperparameter tuning. Thı features, such as OverallQual, GrLivArea, and Neighborhood, that were most correlated with housing prices. The inclusion of these features led to a significa feature engineering.

We tuned key hyperparameters of the Random Forest model to optimize performance. Increasing the number of n_estimators consistently improved the mc increased significantly. This setup achieved an R^2 of 0.90, an RMSE of 27,722.53, and an MAE of 17603.63, indicating strong predictive power and a good fit tc

We also explored the use of PCA to reduce dimensionality and improve computational efficiency. While PCA effectively reduced feature redundancy, it led to model benefits from retaining the full set of high-impact features.

To conclude, Random Forest Regression delivered excellent accuracy across several configurations, with the best R^2 score of 0.90 achieved using a comprehe handling a diverse set of categorical and numerical features effectively.

Support Vector Regression Model

Round 1

Round 1

- Tested values (optimal bolded)
  - C: [0.1, 1, 10, **100**]

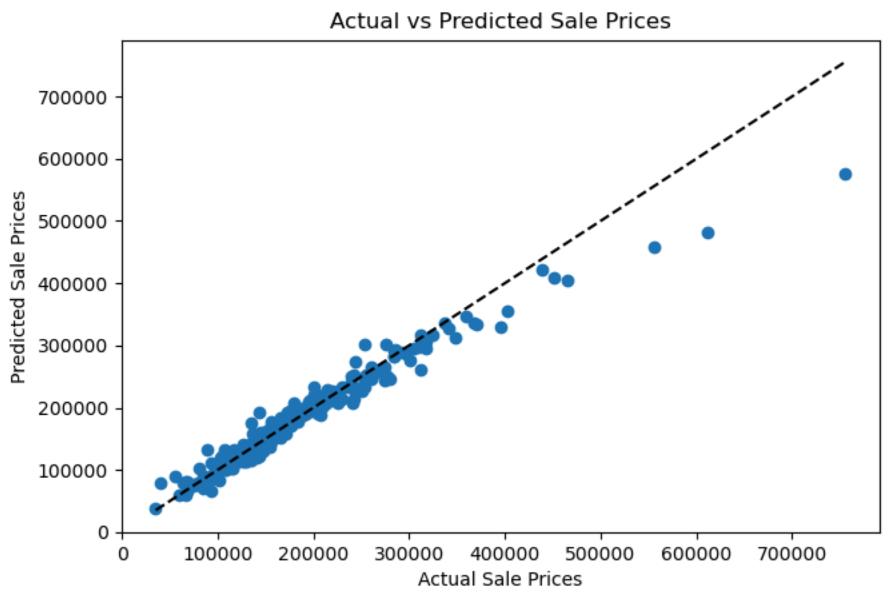
- Gamma: [**0.001**, 0.01, 0.1, 1]

R² Score

0.946

Root Mean Squared Error (RMSE)

20,307.02



Round 2

Round 2

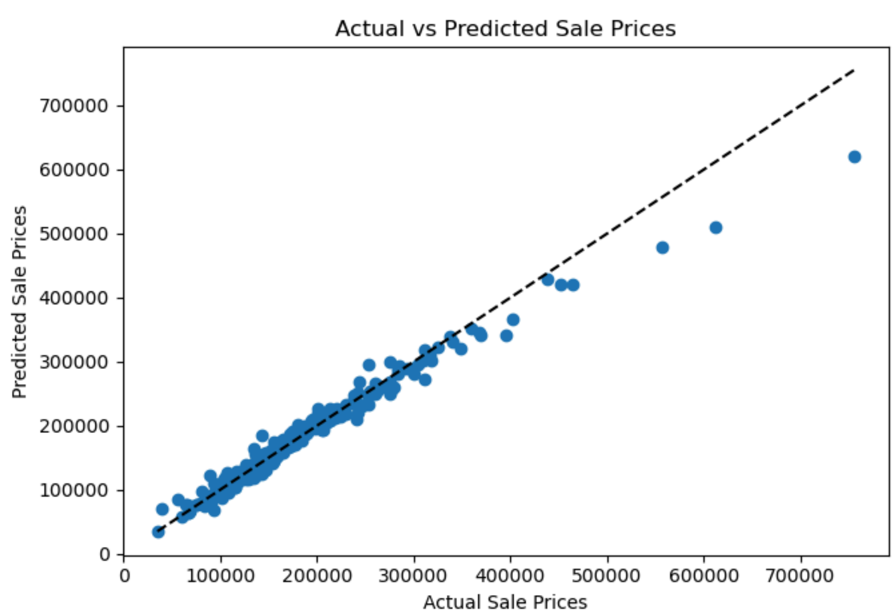
- Tested values (optimal bolded)
  - C: [50, 75, 100, 125, **150**]
  - Gamma: [0.0005, 0.001, **0.005**]

R² Score

0.966

Root Mean Squared Error (RMSE)

9,028.84



Round 3

Round 3

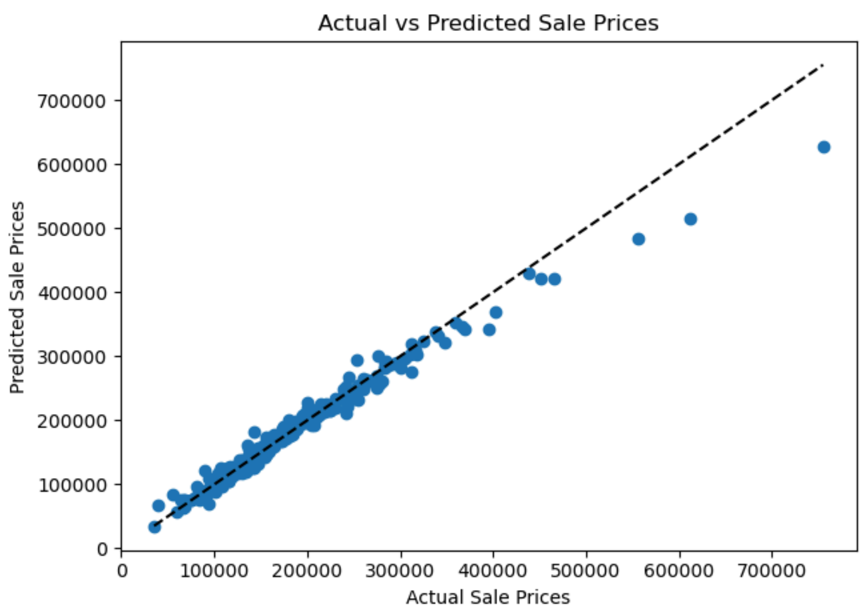
- Tested values (optimal bolded)
  - C: [140, 145, 150, 155, **160**]
  - Gamma: [**0.0002**, 0.0005, 0.0006, 0.0007]

R^2 Score

0.970

Root Mean Squared Error (RMSE)

8,639.46



Round 4

Round 4

- Tested values (optimal bolded)
  - C: [158, 159, 160, 161, **162**]
  - Gamma: [**0.0001**, 0.0002, 0.0003]

R^2 Score

0.971

Root Mean Squared Error (RMSE)

8,548.43

Round 5

Round 5

- Tested values (optimal bolded)
  - C: [160, 200, **400**]
  - Gamma: [**0.0001**]

R^2 Score

0.999

Root Mean Squared Error (RMSE)

733.20

Efficiency Round

Efficiency Round

After 5 rounds of narrowing down, we decided to test performance not just by the accuracy of the model, but by its speed. We tested based on R^2 per seco

- Best R^2/second: 6.171
  - C: 46
  - Gamma: 0.0001

R^2 Score

0.903

Root Mean Squared Error (RMSE)

27,296.69

We tested our Support Vector Regression (SVR) systematically across five rounds of hyperparameter tuning, followed by an efficiency-focused evaluation. Tł (accuracy) and computational efficiency.

First, preliminary testing confirmed that linear was the most suitable one for our dataset; this is likely due to our data's characteristics, which align well with

Next, we moved on to hyperparameter tuning, where through five rounds, we narrowed down C and Gamma to their optimal values. We found that higher va focus more on minimizing prediction error.; however, values too high risk overfitting and require the model to take excessive time to train. Increasing C past values performed better, indicating that a simpler decision boundary was more effective for our data. Lowering Gamma past 0.0001 showed minimal increas

In terms of accuracy, SVR was extremely successful. Round 5 of tuning resulted in an R^2 of 0.99973, an MAE of 733.20, and an RMSE of 1443.90. This means th

After achieving this amazing accuracy, we decided to optimize the model for efficiency by aiming for the lowest R^2 per second of training time. We found tha 0.146 seconds and R^2 of 0.9029. This configuration would be a practical choice when aiming to minimize computational strain while still having decent mode

Final Analysis and Comparison

Analysis of Algorithms/Models

Our analysis began with Linear Regression, which served as an excellent baseline model. Despite its simplicity, it achieved a respectable R^2 of 0.85 and RMSE us to clearly understand the impact of each feature on house prices. However, its assumption of linear relationships between features and prices ultimately l relationships tend to be more complex.

Random Forest emerged as a more versatile solution, achieving an R^2 of 0.90 and RMSE of 27,722.53. The model demonstrated remarkable consistency acro linear relationships in the data. While less interpretable than Linear Regression, Random Forest's robust performance and ability to handle both numerical a effective for our use case.

Support Vector Regression delivered the highest mathematical accuracy with an impressive R^2 of 0.99 and RMSE of 1,443.90. The model excelled at capturin
exceptional accuracy came at the cost of significantly increased computational complexity and training time. The model required careful tuning of hyperpar

Comparison of Models

When comparing these models, several key factors emerged. In terms of raw performance metrics, SVR led the pack, followed by Random Forest, and then L
picture. Linear Regression offered the fastest training and prediction times, with Random Forest following closely behind, while SVR required substantially n

Feature engineering impacted each model differently. Linear Regression benefited significantly from careful feature selection and engineering, requiring ext
flexible, handling raw features well while still improving with thoughtful feature engineering. SVR was only tested with all features, and its performance was
best with a linear kernel, low gamma values, and high C values.

Based on our comprehensive testing, we have two different recommendations. For general house price prediction, we recommend the Random Forest mode
models, its combination of strong predictive performance (R^2 = 0.90), reasonable computational requirements, and robust feature handling makes it a great
prediction or prediction of high value homes, we recommend the SVR model. This model was overall the most accurate and was the only one capable of acc
to the model's ability to capture non-linear relationships and its ability to handle outliers.

Next Steps

Moving forward, there are several avenues for improvement. In terms of model performance, exploring ensemble methods that combine the strengths of mu
maintaining reasonable computational requirements. Implementing cross-validation would provide more robust performance evaluation, while investigatir

Feature engineering remains a promising area for advancement. Developing more sophisticated engineered features, particularly those that capture market
Creating automated feature selection pipelines would streamline the model updating process as new data becomes available.

Data enhancement represents another crucial area for improvement. Incorporating additional relevant features such as school ratings, crime rates, and loca
Expanding the dataset with more recent sales data and incorporating time-series aspects of housing prices would help the model better capture market dyn

References

[1] A. Rawool, D. Rogye, S. Rane, D. Vinayk, and A. Bharadi, “House Price Prediction Using Machine Learning,” | IRE Journals |, vol. 4, pp. 2456–8880, 2021, Ava
[2] J. Guo, “Feature Selection in House Price Prediction,” Highlights in Business, Economics and Management, vol. 21, pp. 746–752, Dec. 2023, doi: https://dc
[3] L. Walthert and F. Sigrist, “Deep Learning for Real Estate Price Prediction,” SSRN Electronic Journal, 2019, doi: https://doi.org/10.2139/ssrn.3393434
[4] Q. Truong, M. Nguyen, H. Dang, B. Mei, “Housing Price Prediction via Improved Machine Learning Techniques”,ScienceDirect,2019, Available: https://www
[5] B. Park and J. K. Bae, “Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data,” Expert Systems
doi:https://doi.org/10.1016/j.eswa.2014.11.040

Gantt Chart

GanttChart.xlsx : Fall

GANTT CHART

https://goo.gl

PROJECT TITLE	Machine Learning Project

TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION
Project Proposal				
Introduction & Background	All	10/2/2024	10/4/2024	2
Problem Definition	Justin	10/2/2024	10/4/2024	2
Methods	Justin,	10/2/2024	10/4/2024	2
Potential Dataset	All	9/30/2024	10/4/2024	4
Literature review	Ayush	10/2/2024	10/4/2024	2
Potential Results & Discussion	Aditya,	10/2/2024	10/4/2024	2
Video Creation & Recording	Justin,	10/3/2024	10/4/2024	1
Presentation	Justin,	10/2/2024	10/4/2024	2
Streamlit Page	Justin,	9/30/2024	10/4/2024	4
Midterm Report				
Model 1 (M1) Design & Selection	All	10/7/2024	10/11/2024	4
M1 Data Cleaning	Vibhav, Ayush	10/11/2024	10/20/2024	9
M1 Data Visualization	Aryan	10/11/2024	10/20/2024	9
M1 Feature Reduction	Vibhav, Ayush	10/11/2024	10/20/2024	9
M1 Implementation & Coding	Justin, Aditya, Ayush	10/11/2024	10/20/2024	9
M1 Results Evaluation	All	10/20/2024	10/23/2024	3
Model 2 (M2) Design & Selection	All	10/7/2024	10/11/2024	4
M2 Data Cleaning	Vibhav, Ayush	10/23/2024	11/1/2024	8
M2 Data Visualization	Justin	10/23/2024	11/1/2024	8
M2 Feature Reduction	Aditya, Justin	10/23/2024	11/1/2024	8
M2 Coding & Implementation	Aditya, Vibhav, Aryan	10/23/2024	11/1/2024	8
M2 Results Evaluation	All	11/1/2024	11/4/2024	3
Midterm Report	All	11/4/2024	11/10/2024	6
Final Report				
Model 3 (M3) Design & Selection	All	11/11/2024	11/14/2024	3
M3 Data Cleaning	Vibhav, Ayush	11/14/2024	11/23/2024	9
M3 Data Visualization	Aryan	11/14/2024	11/23/2024	9
M3 Feature Reduction	Aditya, Aryan	11/14/2024	11/23/2024	9
M3 Implementation & Coding	Justin, Ayush, Vibhav	11/14/2024	11/23/2024	9
M3 Results Evaluation	All	11/14/2024	11/23/2024	9
M1-M3 Comparison	Aryan, Aditya, Justin	11/23/2024	11/26/2024	3
Video Creation & Recording	All	11/23/2024	12/3/2024	10
Final Report	All	11/23/2024	12/3/2024	10

PHASE ONE															
Sep 16							Sep 23								
M	T	W	R	F	S	U	M	T	W	R	F	S	U	M	T

Fall

Contribution Table

Contribution Tables

Project Proposal | Midterm Checkpoint | **Final Report**

Ayush Sharma	Conducted testing and optimization of the Random Forest model, systematically evaluating feature sets, and hyperparameter configurations to maximize accuracy. Explored the impact using PCA and identified the best-performing configuration, achieving an R^2 of 0.90.
Aditya Kabu	Performed testing and fine-tuning of the Random Forest model by assessing various feature sets and hyperparameter combinations to enhance accuracy. Investigated the role of PCA in feature selection and identified an optimized configuration that delivered strong predictive performance.
Aryan Verma	Collaborated with team members to incorporate domain knowledge into model design and testing. Created the final presentation and Video highlighting and combining all parts of the project. Evaluated trade off with different model to come up with the best results.
Vibhav Agrawal	Worked with team to design and source training and testing data. Did testing with different parameter combinations to improve results of varying models helping identify best configurations for each method.Edited and recorded final video powerpoint presentation elaborating on project details, methods and conclusions