# House Price Regression

Mark Herrmann – Team Leader, Algorithm Design

Joseph Hooser – Feature and Label Analysis, Preprocessing

Jason Powell – Implementation

Johntae Leary – Documentation

# Project Overview

**Objective:** Predict housing prices based on property features

**Significance:** To help buyers, sellers, and realtors make data-driven decisions

**Approach:**

- ❖ Train multiple regression models trained on cleaned housing dataset
- ❖ Tune Hyperparameters for each regression model using GridSearchCV
- ❖ Identify top performer and analyze results

# Literature Review

**Paper:** Ni Chuhan's “House price prediction based on different models of machine learning"(March 2024)

**Significance:**

- The study compares the performance of these 4 common regression algorithms on a house price prediction task using a Kaggle dataset.

**Table 4.** Results of models.

| Model | MAE | MSE | RMSE | $R^2$ score |
|---|---|---|---|---|
| Linear Regression | 23567.89 | 1414931404.63 | 37615.57 | 0.8155 |
| Support Vector Machines | 17843.16 | 1132136370.34 | 33647.23 | 0.8524 |
| Random Forest Regressor | 18040.67 | 950844232.54 | 30835.76 | 0.8760 |
| XGBoost | 19411.23 | 900813561.35 | 30013.56 | 0.8826 |

**Contribution:** This paper allows us to identify regressors that performed well on housing datasets(i.e, XGBoost, Random Forest), and what error metrics we can shoot for.

# Dataset Overview

- **Source:** Kaggle “House Prices: Advanced Regression Techniques”
- **Training Set:** 1,460 Rows, 81 Features
- **Validation Set:** 20% of Training Set
- **Target Variable:** SalePrice

- **Notable Features**
  - **Numerical:** LotArea, GrLivArea,TotalBsmtSF
  - **Categorical:** Neighborhood, HouseStyle, GarageType

# Feature Engineering

We added some Features to better capture the data movements

Total Square Feet

- 'TotalSF' = 'GrLivArea'+ 'TotalBsmtSF'

Age of House

- 'HouseAge' = 'YearBuilt' - 'YrSold'

Age of a Remodel

- 'RemodelAge' = 'YearRemodAdd' - ‘YrSold’

*GrLivArea’ is the total square Feet above ground

# Handling Missing Values and Feature Encoding

- **Numerical Fixes:**
  - LotFrontage: filled using neighborhood median
  - GarageYrBlt: Values were missing so we assumed it was built the same year the house was
- **Categorical Values:**
  - Electrical: One house was missing this so we filled it with the most common value
  - Other features like fences that were left null were filled with none
- **Encoding:**
  - One-Hot Encoding expanded the feature list from 81 to over 150, and allowed us to use categorical variables.
- **Scaling:**
  - StandardScaler was applied to numerical features
- **Data Split:**
  - 80% of the Data was used for training
  - 20% was used for validation

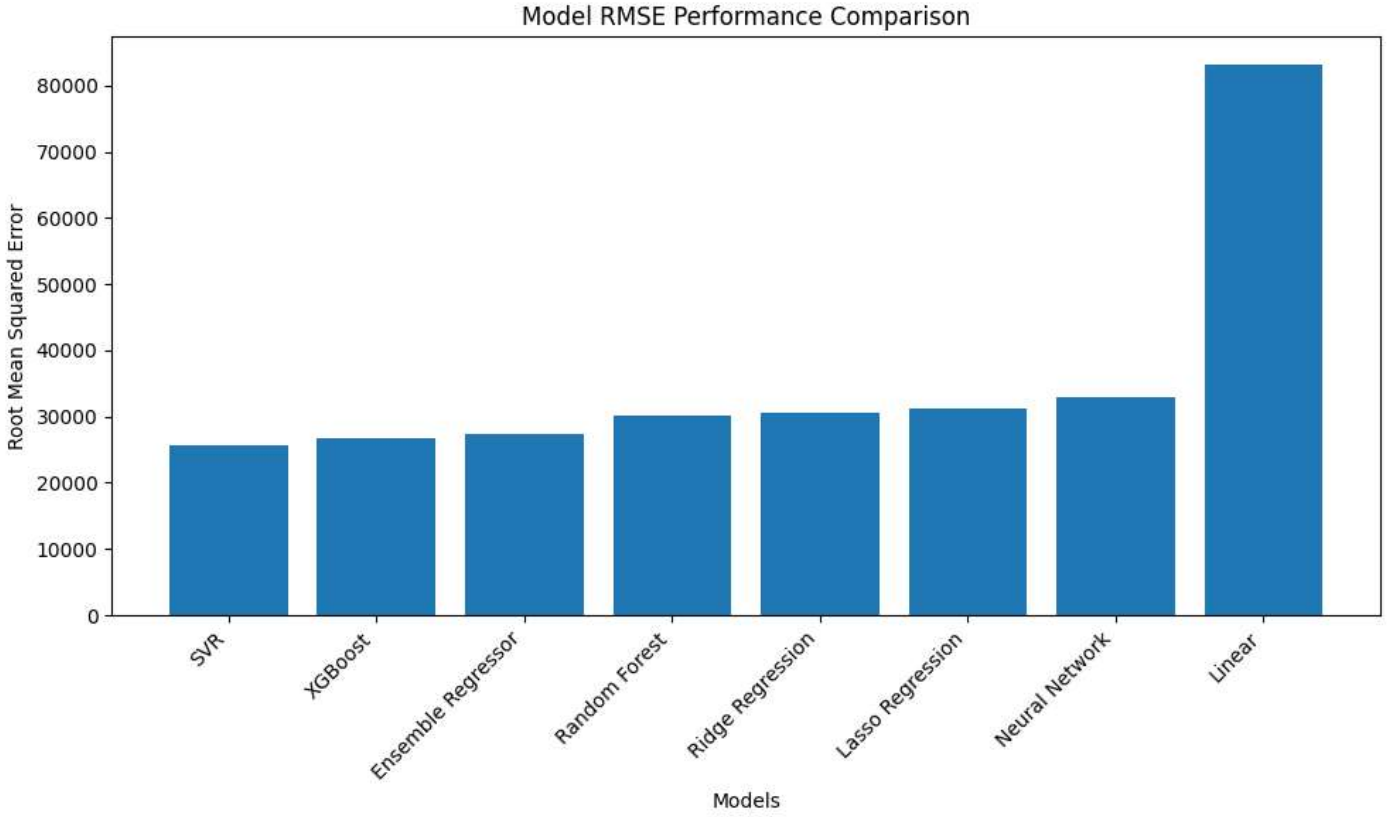
# Model Evaluation and Comparison

| Model | RMSE | R-Squared | MAE | Time |
|---|---|---|---|---|
| SVR | 25644 | 0.914 | 14830 | 1.0 |
| XGBoost | 26753 | 0.907 | 16587 | 15.06 |
| Ensemble Regressor | 27376 | 0.902 | 15844 | 178.57 |
| Random Forest | 30089 | 0.882 | 17928 | 9.09 |
| Ridge Regression | 30604 | 0.878 | 18957 | 0.39 |
| Lasso Regression | 31256 | 0.873 | 18770 | 0.41 |
| Neural Network | 32851 | 0.859 | 20205 | 152.63 |
| Linear | 83215 | 0.097 | 23965 | 1.2 |

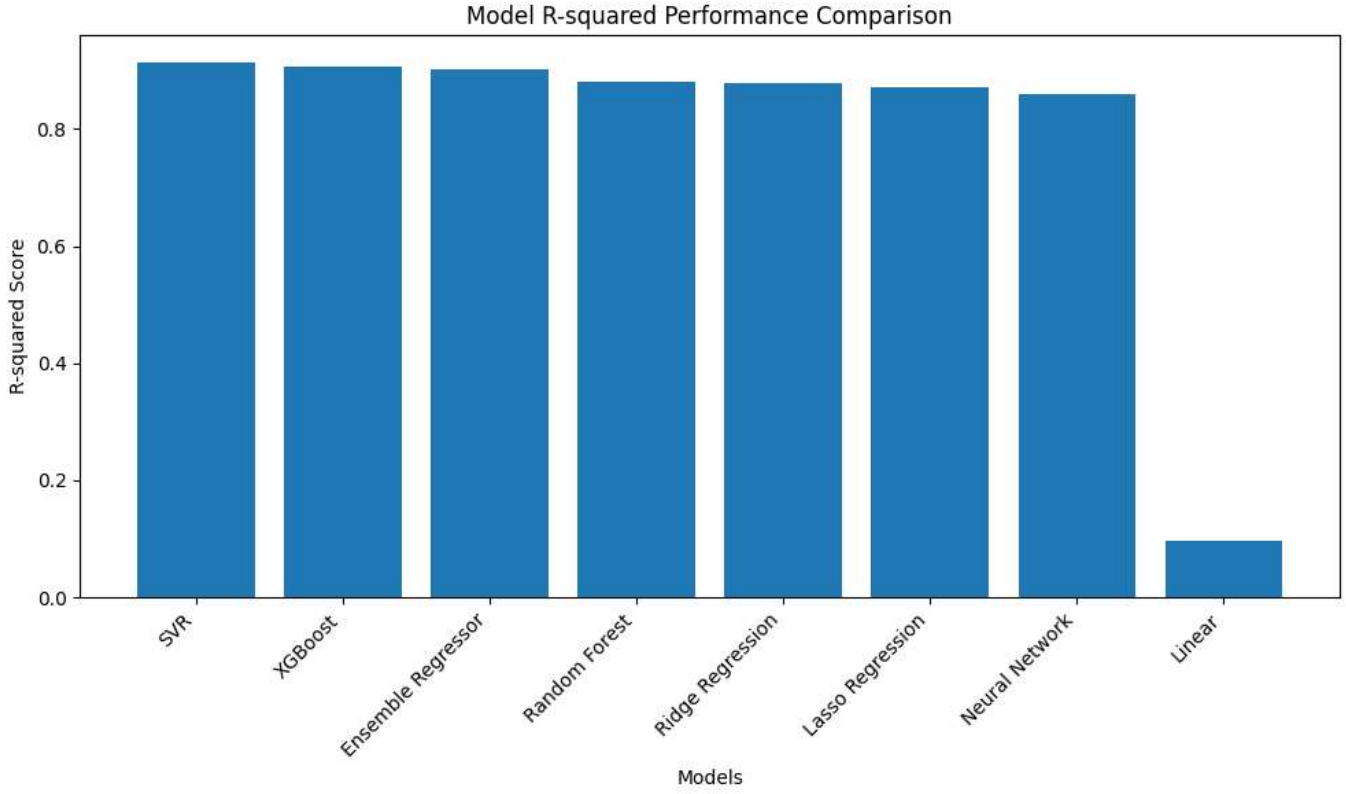
**Results:**

- SVR performed the best and also had a very fast training time compared to the other top performers (XGBoost and Random Forest)
- Each Model had its hyperparameters tuned using GridSearchCV
- The Ensemble Method:
  - The average of our top 6 regressors also performed very well, however models need to be trained - computationally intensive

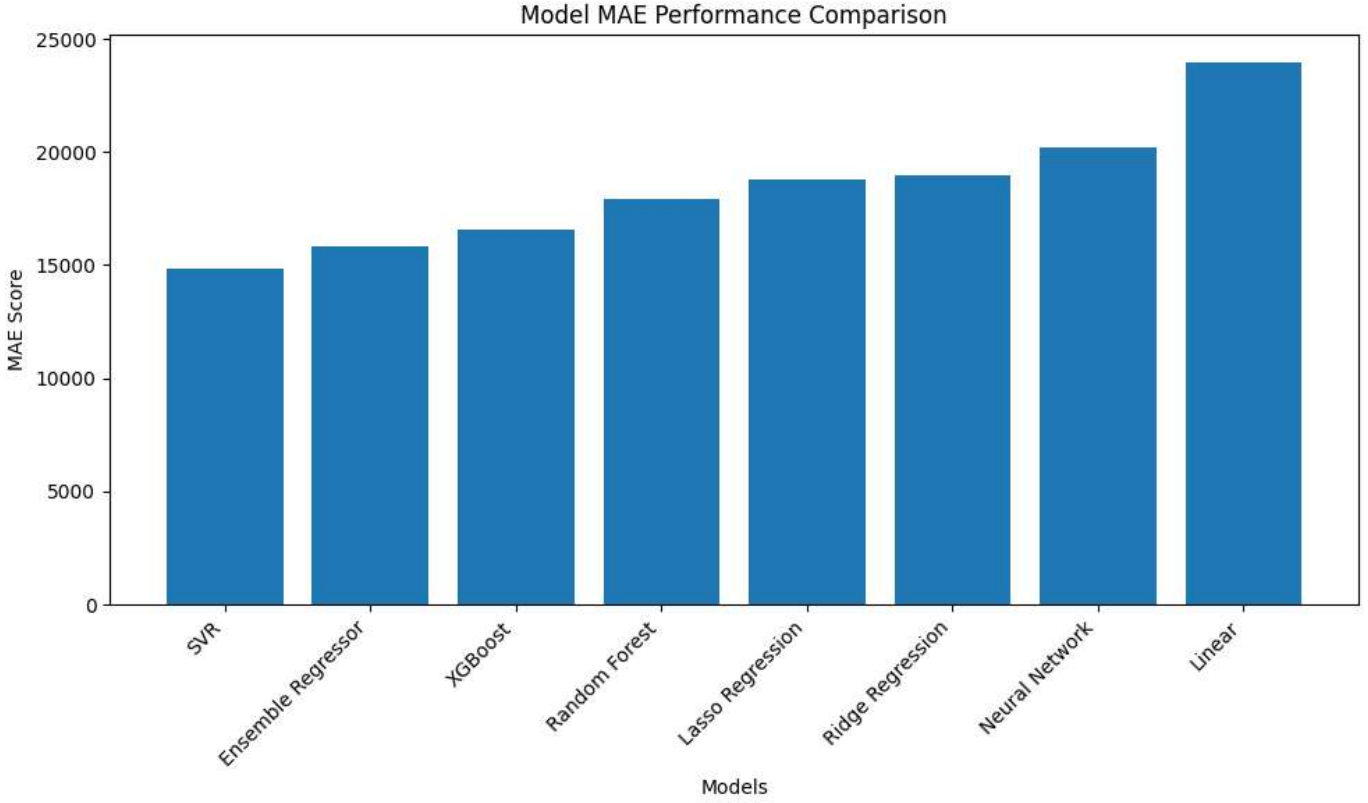
# Root Mean Squared Error Comparison



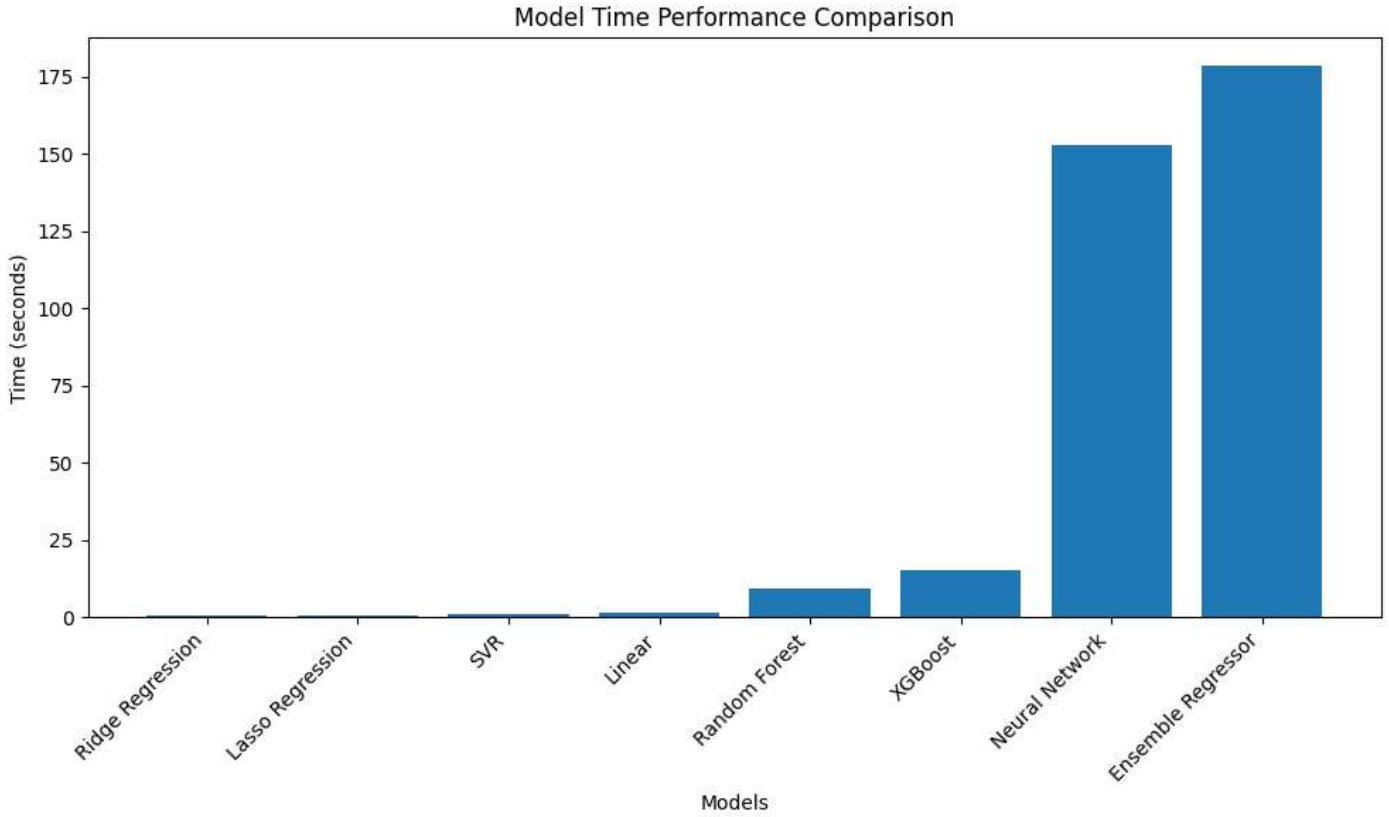
# Model's R-Squared Comparison



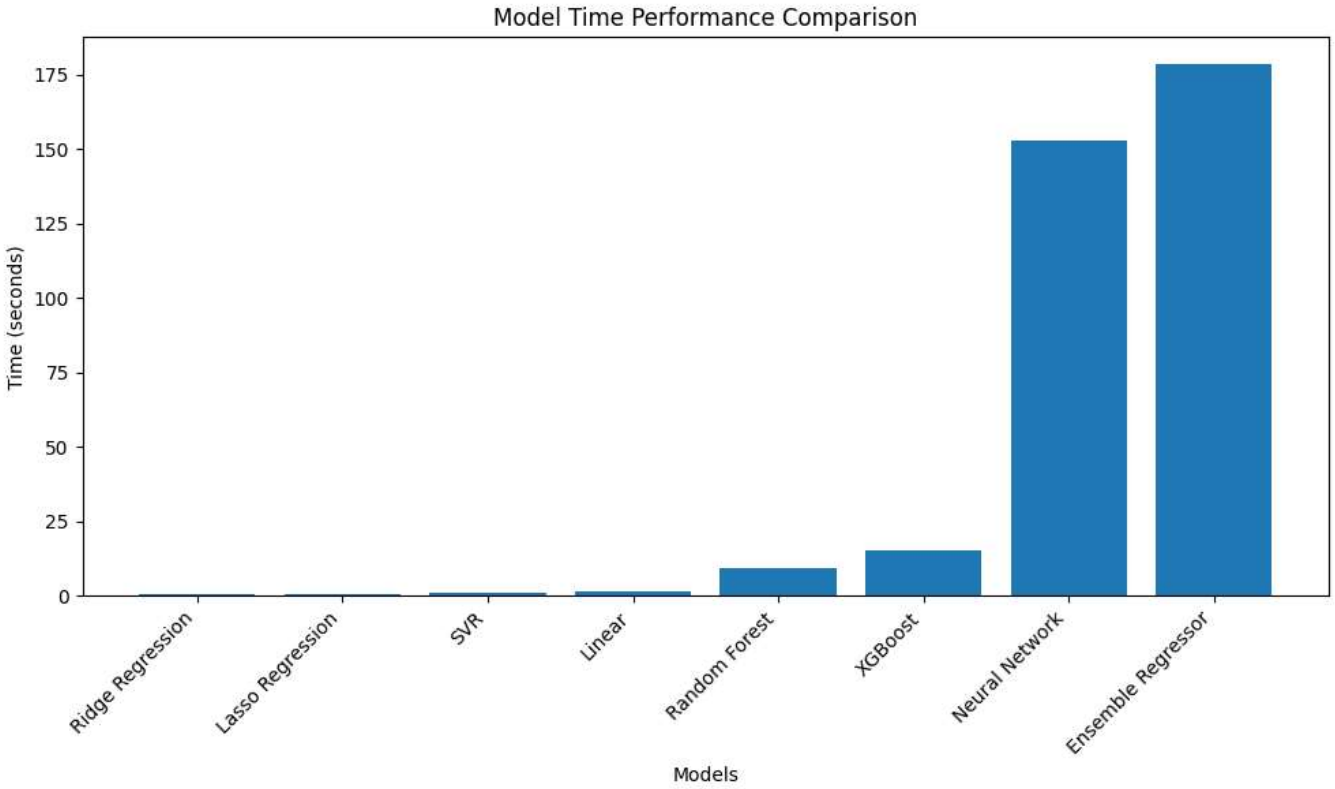
# Mean Absolute Error Comparison



# Training Time Comparison



# Model's Training Time Visualized



# Cross Validation Results for SVR

- **Objective:** Get a more reliable estimate of SVR's performance and variability across different subsets of the training data

| Folds | Mean RMSE | Mean R-Squared | Mean Absolute Error | Runtime (seconds) |
|---|---|---|---|---|
| 20 | 26,749 | 0.823 | 15,688 | 13 |
| 50 | 24,362 | 0.774 | 15,510 | 34 |
| 200 | 20,788 | 0.665 | 15,167 | 157 |

- **Conclusion:** As you can see, the more folds that the dataset is trained on, the Mean RMSE and MAE improve, however the R-Squared value gets worse - meaning less of the variance is getting captured by the model. This could mean a model with more bias may perform better

# Model Limitations and Considerations

**Assumptions**:

- The Regressor assumes future data follows similar patterns
- The housing market can be volatile
- Our model doesn’t account for macroeconomic shifts or interest rate changes

**Dataset:**

- The houses are all from Ames, Iowa, and would likely only perform well on other houses within this area

**Outliers:**

- May underperform on rare or unusual house types
- Overfitting risk with unbalanced categories, may need to introduce more bias

# Contributions to Existing Studies

- **Improved upon Studies results:** We were able to find a better performing model for predicting house prices - Support Vector Regression.
- **Better tuned Parameters:**
  - We were also able to better tune SVR's hyperparameters in order to deliver better results than the study was
  - Hyperparameters:
    - 'kernel': ['poly'],
    - 'C': [1000],
    - 'epsilon': [1000],
    - 'gamma': ['scale'],
    - 'degree': [3],
    - 'coef0': [3]
- The Literature's lowest RMSE was 30,013 and we achieved an RMSE of 25,644

# Future Development

**Further Hyperparameter Tuning:** More extensive tuning of the best models (SVR, XGBoost, Random Forest) with more exact hyperparameters.

**Advanced Feature Engineering:** Explore interactions between more features and explore external data sources (e.g., economic indicators, local development plans).

**Outlier Analysis**: Further analysis and handling of potential outliers in features or target variable.

**Deployment**: Develop a system to deploy the trained model for making predictions on new

# Conclusion

- **Summary:** Successfully implemented, tuned and evaluated multiple regression models for house price prediction
- **Key Findings:**
  - Data preprocessing and feature engineering were critical steps.
  - Ensemble methods (XGBoost, Random Forest) significantly outperformed the linear regressor, but they had longer training times
  - SVR achieved the best performance based on validation and cross-validation metrics (lowest RMSE, highest R2).
- **Project Success:** Delivered a predictive model capable of estimating house prices with high accuracy within ~ $20,688 (as measured by RMSE/R2).

# Questions?