May Mon Thant

MMDT MLAI101

Midterm Project Report

Instructor: Dr.Myo Thida

Group Mentor: Ma Nuwai Thet

Date: July 19, 2025

### **Problem Description**

In this project, I am trying to predict the house price data. I did a similar one in the class but this time a more complex dataset with many different features. I chose this problem because I think this is the real world problem where we can use this model in predicting real house prices.

### **Dataset**

The Dataset is from Kaggle. I choose the dataset from house price predicting competition. And there are many columns which I can use as features. There are altogether 80 columns and some of them contain categorical data such as . condition and also date and time data. Among them, I chose the columns which can best predict the sale price which are

'LotArea', 'TotalBsmtSF','GrLivArea','FullBath','YearBuilt','YrSold','1stFlrSF'.

| **Column Name** | **Meaning** |
| --- | --- |
| LotArea | Size of the property lot |
| TotalBsmtSF | Total basement square footage |
| GrLivArea | Above-ground living area |
| FullBath | Number of full bathrooms |
| YearBuilt | Construction year of the house |
| YrSold | Year the house was sold |
| 1stFlrSF | First floor square footage |

For the train dataset, I chose numerical data without missing values. I choose this column because I think people want to check the area of the house, year built , year sold and total square feet. Then I had 1,168 training samples and 292 test samples, each with 7 features. That’s why before I build the model, I consider adding them as features.

### **Modeling Approach**

I first initiated data cleaning, checking missing values in the columns that I have chosen as features and then I checked for the statistics of the dataset. Then, I add StandardSaler to normalize input of the data as I also intend to test with polynomial features. Before starting the modeling, I split the data to train and test so that I can test my train data with unseen data and can know how my model works well with new data.

**Multiple Linear Regression**

First of all, I started with multilinear regression as I have added a total of 7 features and my target variable as sale price. I guess my multi linear regression model also works well in training and testing with data.

**Polynomial Regression**

After that I tried polynomial regression, with the same features. I tested different degrees to see which degree can be the best. In this case, I use The grid search method to find the hyper parameter order of the polynomial regression model. Pipeline function is used to avoid the potential data leakage issue. So I found out that degree 2 is the best for the model.

### **Evaluation**

To choose the model which can better perform please reference the table.

Metrics

| Metrics | Multilinear Regression | Polynomial Regression |
| --- | --- | --- |
| R^2 | 0.7379(73.8%) | 0.8017 (80.2%) |
| MAE | 28.059 | 25,192 |
| MSE | 2,010,646,852 | 1,521,378,000 |

Multiple Linear Regression Model

| Dataset | R^2 Score | MSE | MAE |
| --- | --- | --- | --- |
| Training Set | 0.686 (68.6%) | 1,873,843,017 | 26,661 |
| Test set | 0.738 (73.8%) | 2,010,646,852 | 28,059 |
| Difference | Around 5.2% greater on test | Higher on test | Higher on test |

Polynomial Regression Model

| Dataset | R^2 Score | MSE | MAE |
| --- | --- | --- | --- |
| Training Set | 0.797 (79.7%) | 1,213,496,000 | 23,967 |
| Test set | 0.802 (80.2%) | 1,521,378,000 | 25,192 |
| Difference | Around 0.5 % greater on test | Higher on test | Higher on test |

Cross Validation check

| Model | R^2 |
| --- | --- |
| Multi Linear Regression | 0.685 (68.5%) |
| Polynomial Regression | 0.680 (68.0%) |
| Difference | 0.5 difference |

If we compare the two models, 80.2% vs 73.8% the polynomial regression with degree 2 performs better. And also the results of MAE and MSE are less than in Polynomial rather than in multiple linear regression.

If we check on train and test performance of both models. Linear Regression shows that 68.6% R² on training, 73.8% R² on test and then Polynomial regression shows that 79.7% R² on training, 80.2% R² on test. This is where my model can perform better on test results than train one.

That’s why I performed a cross-validation check. The results of cross-validation shows that while polynomial regression performs better on a single test split, the two models are extremely similar (~68% R²) when tested seriously on many data splits.

After checking the degree 2 training and test results, I think this is the choice for the model and then I chose the polynomial model which I think can be relevant more with real world predictions of data. This is because it stayed at the highest level of performance on the held-out test set and captured non-linear effects that are easy to understand in real estate prices.

### **Reflection**

The challenges are that firstly I had difficulties in choosing the dataset. This is because there are many datasets and it is hard to choose the right one. So, I challenged myself to choose the ones with complex columns.Then, I had no idea what features I should use. So, I researched some factors that might influence the house pricing and then I balanced my features with real world factors and non-missing values data. In my case, both models perform well so it is difficult to chooseThe big difference in performance (80.2% vs 73.8%) with the very small cross-validation difference (68.0% vs 68.5%).It seems very unlikely that my model fits my test data better than my training data.

Feature selection from 80 available columns required domain knowledge about real estate markets. . Since there can be many columns influencing the sale price of the house, I was not able to add the location and other categorical variables.

Next time, I will spend time on choosing other features and then also try to deploy my model for better test experience. In this project my model can predict the accuracy of 68-80%.