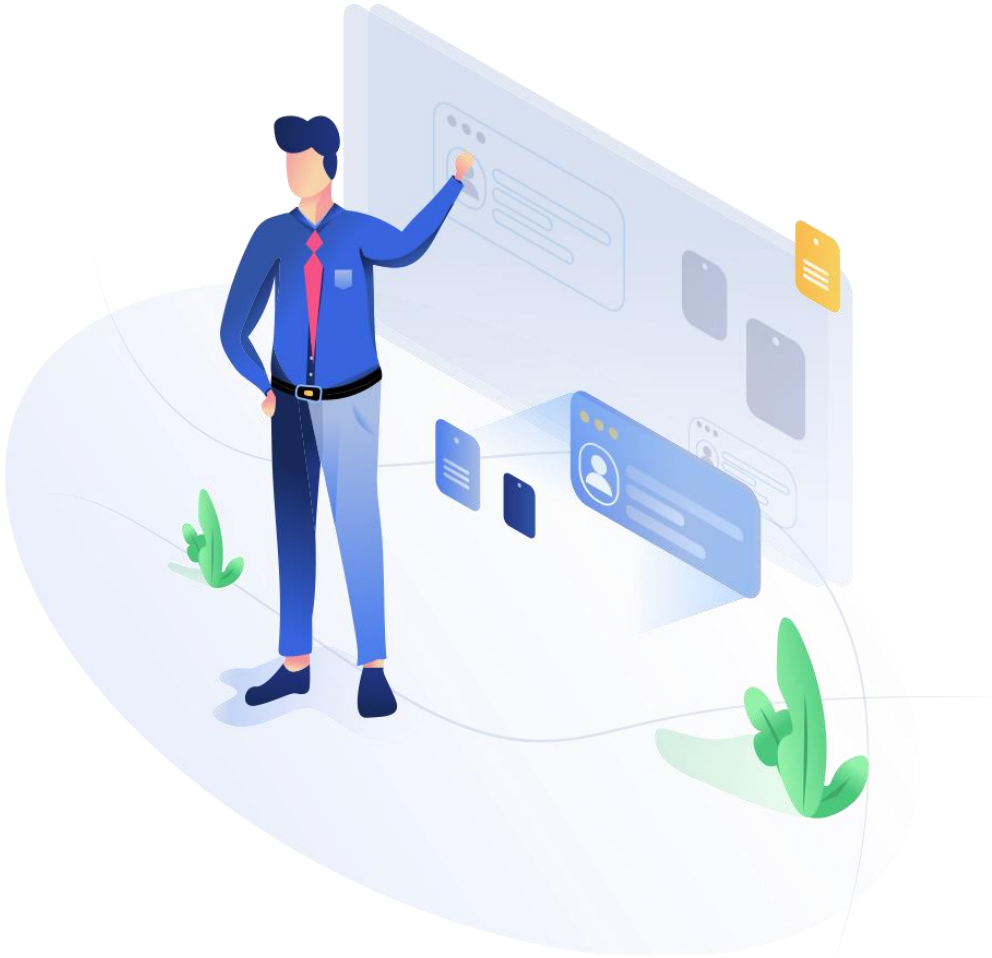
Housing Data Predictions Linear Regression and KNN

J a s o n T a n g

03/18/2023

C O N T E N T S

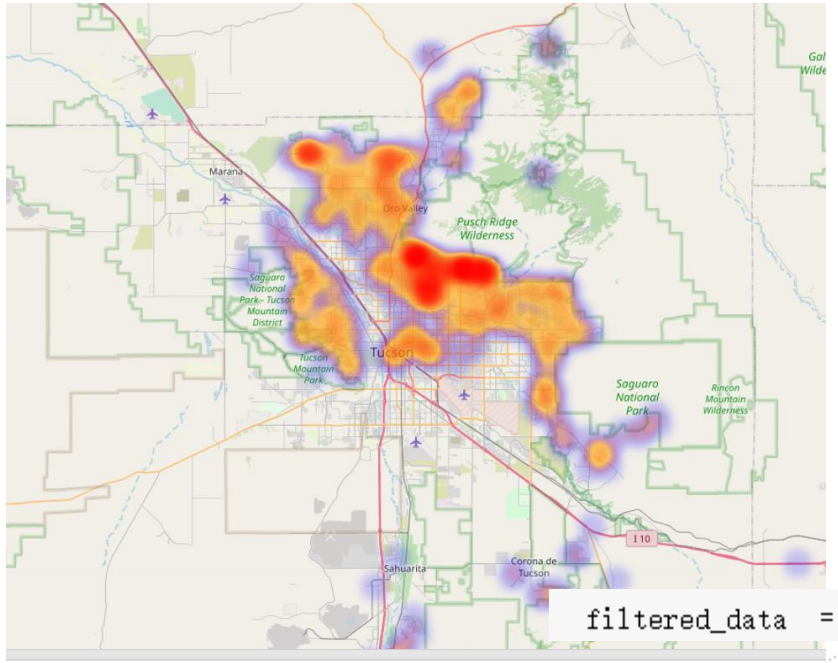
01 PART ONE Luxury Retail development



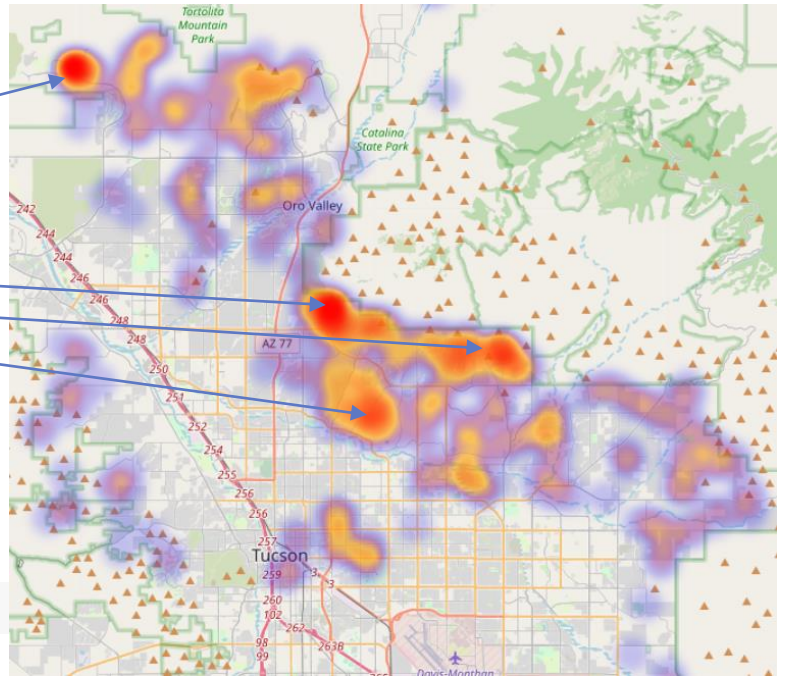
Luxury retail development is an important part of the economy, but making informed decisions about where and how to develop luxury retail spaces can be challenging. In this project, we are using data to help inform these decisions

To achieve this goal, we will be using a dataset that contains information on luxury retail properties in a major city(Arizona), including the sold price, longitude, and latitude of each property. Our analysis will focus on using these variables to identify areas that are likely to have high demand for luxury retail development

Dataset we will be using contains 5000 Rows of data and contains 16 features, we have cleaned the data and kept what is valuable to our analysis and our model.



Potential areas for Luxury retail development



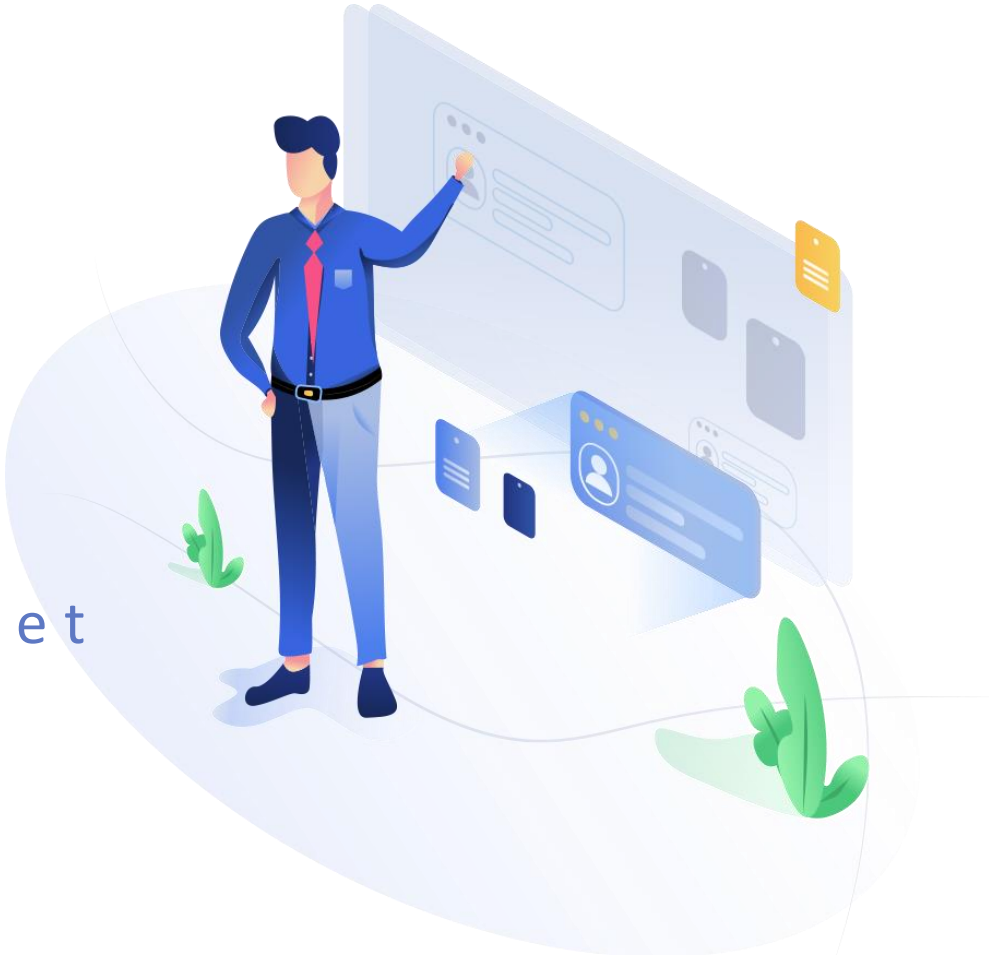

```
filtered_data = house_data[house_data['sold_price'] > 800000]
```

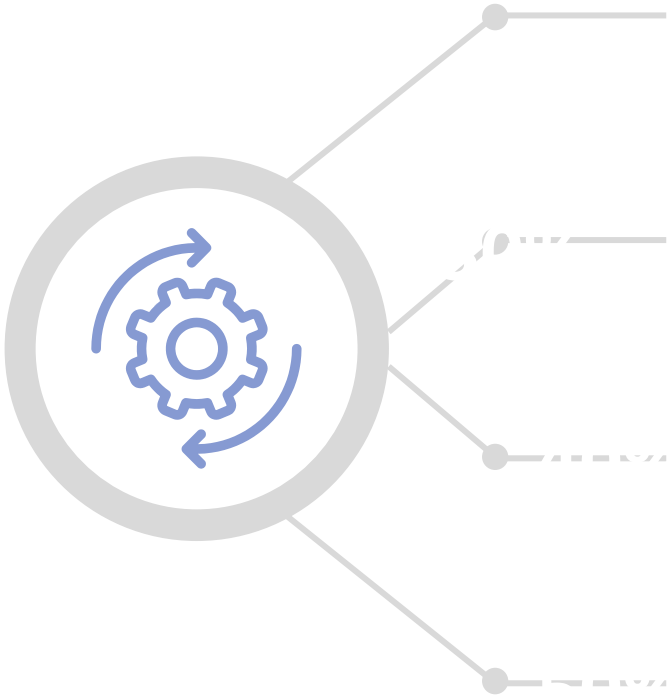
Most Houses are in Tucson, Arizona.

We can see that most of the high value houses are not so close to the center of Tucson.

02 PART TWO

Prepare the Dataset

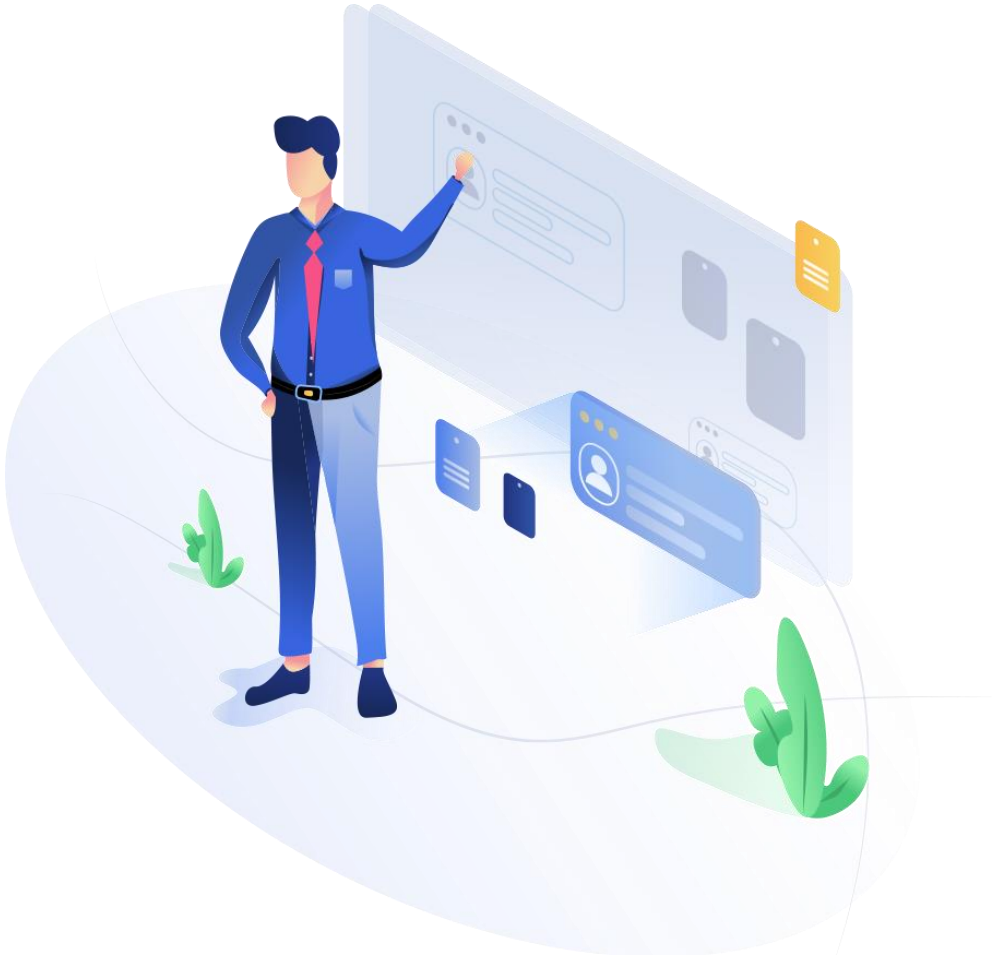




- Our house dataset is in CSV format and has been cleaned.
- Data cleaning: dropping unnecessary 0s and Nones, select the features has strong correlation to sold price, and important features that is relevant to use case
- Check the shape and transfer it to array using numpy
- Now we can use algorithms

03 PART THREE

KNN(k- nearest neighbors)



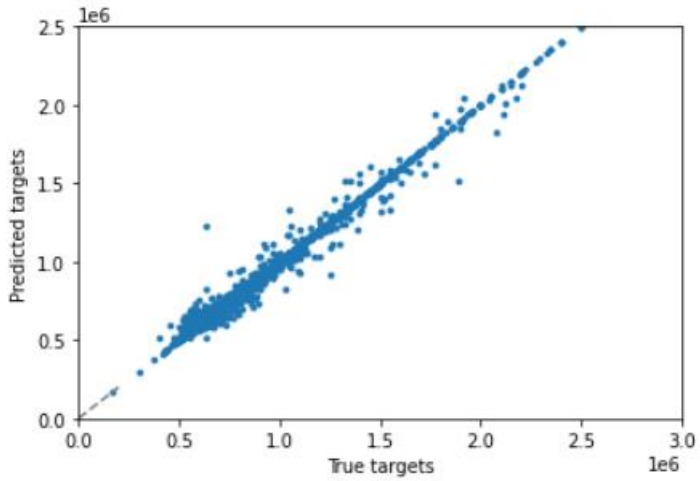
KNN can be particularly useful for house data, where there may be many features that can impact the price of a house

The KNN Classifier is a non-parametric algorithm that works by finding the k closest images in the training set to the input image

KNN with K=5

```
X = df[['sold_price', 'longitude', 'latitude','year_built','bedrooms','sqrt_ft','lot_acres','fireplaces']].copy()
X = X.to_numpy()
y = X[:, 0]
X = X[:, 1:]
```

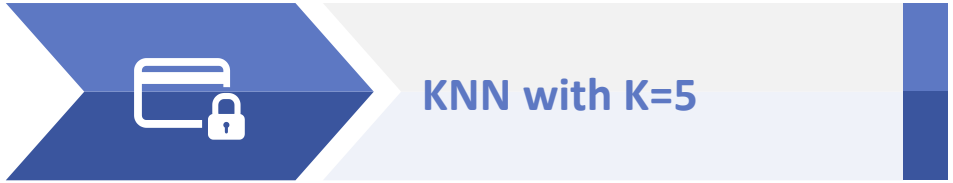
Here I have selected the features that's has the strongest correlation and are important to our use case



Fit line close to diagonal.

After I trained the model and evaluated the model, I got 0.99 R – squared value.

```
R-squared value: 0.9928901739547791
```



KNN with K=5

```
[ ]  mse = mean_squared_error(y, y_pred)
     print("MSE:", mse)
```

```
MSE: 710047357.2301424
```

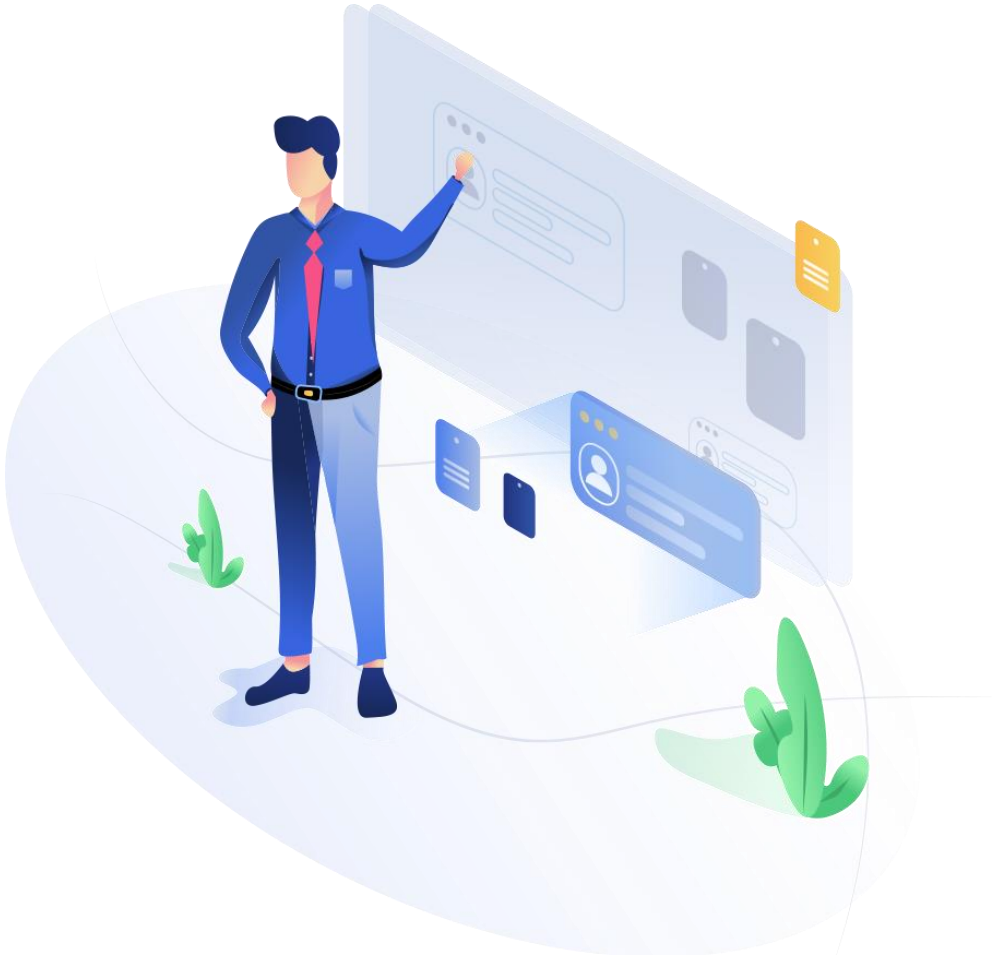
```
print("Accuracy:", accuracy(y,y_pred))
```

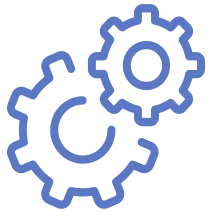
```
Accuracy: 0.2111336032388664
```

This is suggesting the model is overfitting
and doesn't perform really well

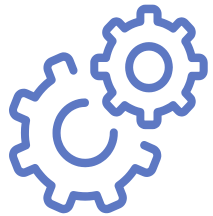
04 PART FOUR

Linear regression



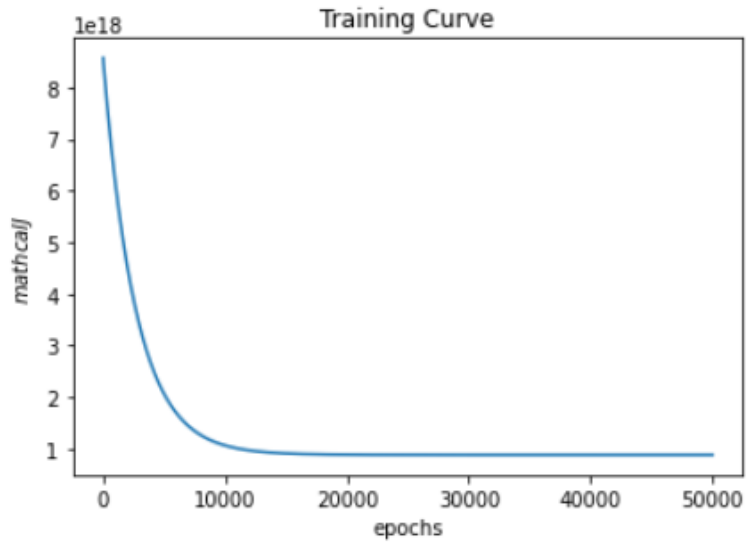


Linear regression is a statistical model used to examine the relationship between a dependent variable and one or more independent variables

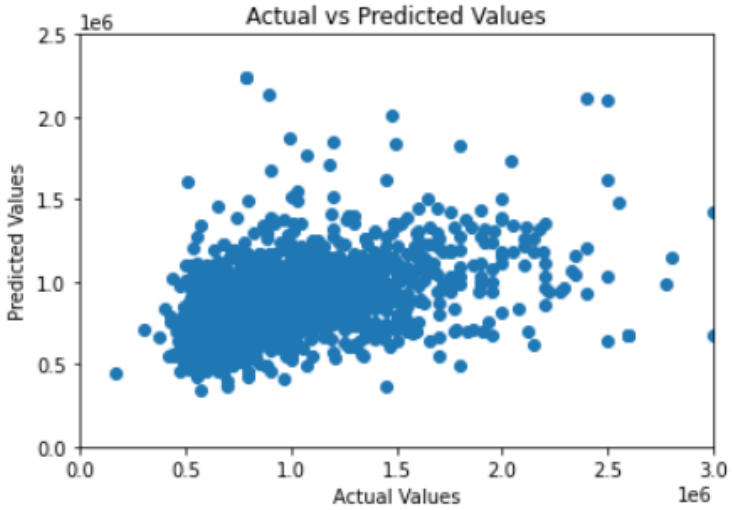


Linear Regression Model

```
myReg.fit(X, y, epochs=5e4, eta=1e-11, show_curve=True)
```



This shows us our model is not overfitting




```
R-squared value: 0.2804605000026187
```

Diagonal fit line

Linear Regression Model

```
X=df[['sold_price','taxes','year_built','bedrooms','sqrt_ft','garage','fireplaces']].copy()
```

```
Xtest=np.array([[3000,1998,3,3000,2,2]])
```

```
myReg.predict(Xtest)
```

```
array([657354.60465992])
```

```
my_r2_score = r2_score()
score = my_r2_score.R2(y, y_pred)
print("R-squared value:", score)
```

```
R-squared value: 0.28016828832373064
```

Usable model and prediction is respectable

Conclusion

01 Advantage of using KNN on house data is its simplicity. KNN does not require any assumptions about the underlying distribution of the data and can be used for both linear and nonlinear relationships between the features and the target variable

02 Our house data, where the relationship between features (such as square footage, number of bedrooms, and location) and the target variable (such as house price) may be nonlinear

03 KNN Did not achieve a good performance on this house dataset compared to Linear regression which has good and respectful performance

Thank you