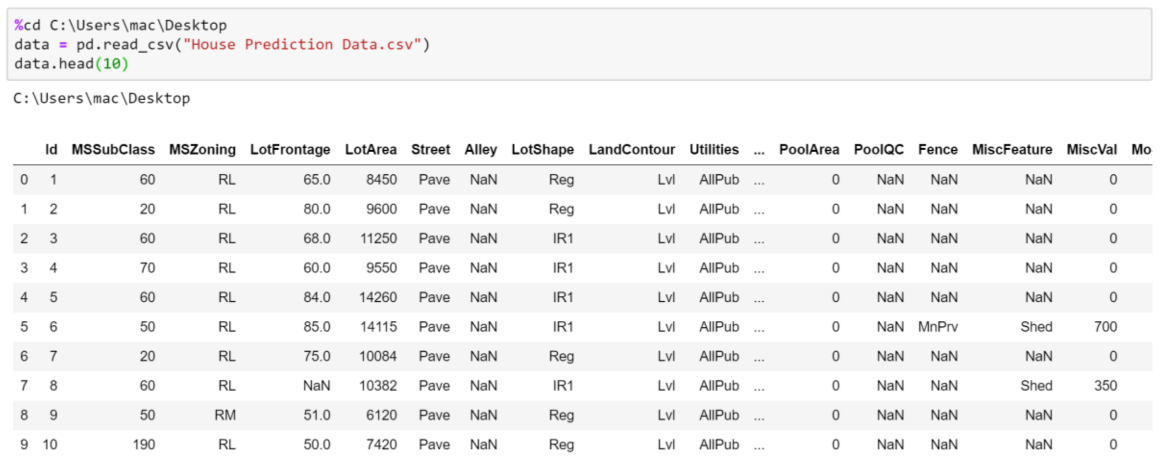
Project 1 house price:

The main goal we want to realize is to predict the housing price, and the ultimate service we could provide is that when customer provide some data to us, we can predict the house price and give objective suggestion for customers.

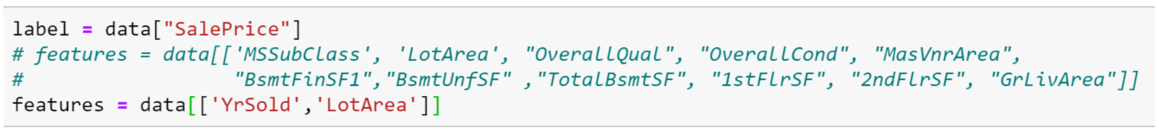
Firstly, the csv file, House Prediction Data.csv, should be loaded. Using head function to observe what happened and what kind of data can be used to do the regression. After that I choose some of the column to do deeper research. So I choose sale price to be the label, which is what we want to predict, and choose year sold and lot area to be the features that can be analyzed, based on the experience.



```
%cd C:\Users\mac\Desktop
data = pd.read_csv("House Prediction Data.csv")
data.head(10)
```

```
C:\Users\mac\Desktop
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	Mo
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
5	6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	MnPrv	Shed	700	
6	7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
7	8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	Shed	350	
8	9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
9	10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

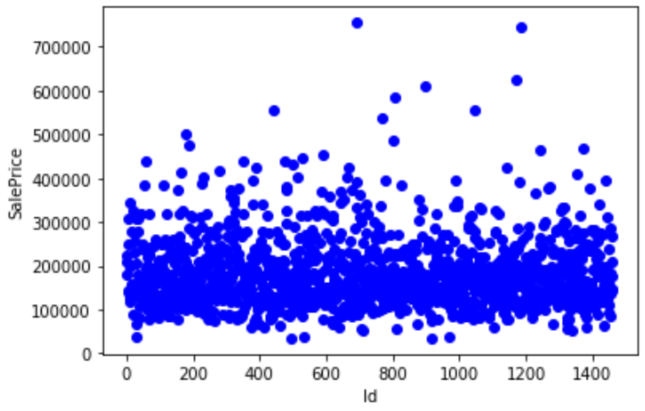


```
label = data["SalePrice"]
# features = data[['MSSubClass', 'LotArea', "OverallQual", "OverallCond", "MasVnrArea",
#                  "BsmtFinSF1","BsmtUnfSF" ,"TotalBsmtSF", "1stFlrSF", "2ndFlrSF", "GrLivArea"]]
features = data[['YrSold','LotArea']]
```

Then I use three visualization method to observe the relationship between these parameters and features. The first scatter plot illustrates the relationship between ID and Sale Price. On the overall view, there is not an obvious relationship between these two parameters, which means ID could not influence the sale price, and also fits what I imagine. When we come to details, the ID of the highest sale price is 760 approximately, and that of lowest is around 920.

```
plt.figure()
plt.scatter(data["Id"], label, color = "blue")
plt.xlabel("Id")
plt.ylabel("SalePrice")
```

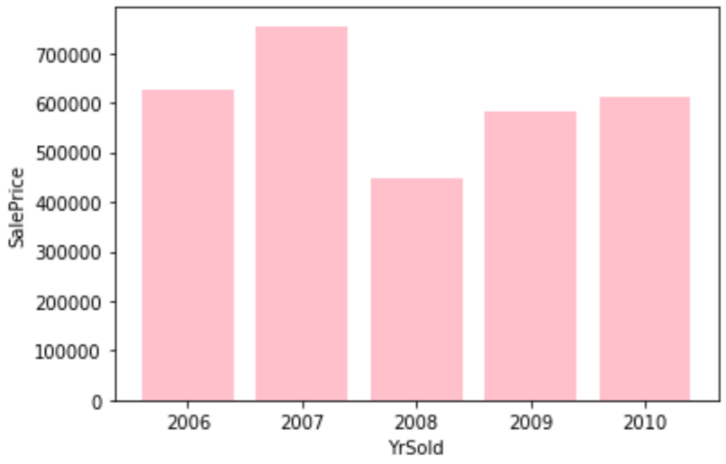
```
Text(0, 0.5, 'SalePrice')
```



The second bar plot shows the relationship between year sold and sale price. Based on the plot, we can obtain that the highest sale price is in 2007, while the lowest is in 2008. From 2008 to 2010, there is a steady increase and reach to the around 600000 in 2010, but still lower than 2008's.

```
plt.figure()
plt.bar(data["YrSold"], label,  color = "pink")
plt.xlabel("YrSold")
plt.ylabel("SalePrice")
```

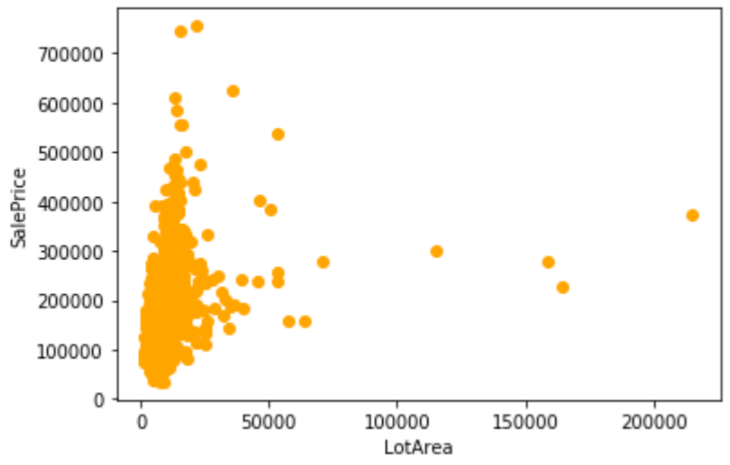
```
Text(0, 0.5, 'SalePrice')
```



Last I want to figure out the relationship between the lot area and sale price. On account of the scatter plot, most of the lot area lay between 0 to 50000, and the sale price lay from 20000 to 500000.

```
plt.figure()
plt.scatter(data["LotArea"], label,  color = "orange")
plt.xlabel("LotArea")
plt.ylabel("SalePrice")
```

```
Text(0, 0.5, 'SalePrice')
```



Visualization analysis is not as accurate as data analysis. So I choose to calculate the correlation between these two parameters to see whether there is a relationship between themselves. As we can see, the correlation between year sold and lot area is -0.024, while the correlation between year sold and overall quality is -0.019. We can make a relative speculate that the relationship between year sold and overall quality is lit bit stronger than that between year sold and lot area.

```
print("correlations of 'YrSold','LotArea' is ", np.corrcoef(data['YrSold'],data['LotArea'] ))
print("correlations of 'YrSold','OverallQual' is ", np.corrcoef(data['YrSold'],data['OverallQual'] ))
```

```
correlations of 'YrSold','LotArea' is  [[ 1.         -0.02423447]
 [-0.02423447  1.        ]]
correlations of 'YrSold','OverallQual' is  [[ 1.         -0.01961377]
 [-0.01961377  1.        ]]
```

For the extension, I do all of them.

At the beginning, I divide the data in to two sets, one is training set and the other is validation set. The goal why I do so is to make sure the model will not overfitting to training data. In liner regression, the mean squared error is 5783737855, which is extremely great. I have one reasonable speculate, the first is that linear regression itself cannot support too much fluctuation, so the forecast cannot be accurate.

```
features = preprocessing.scale(features)
X_train,X_test, y_train, y_test = train_test_split(features, label, test_size=0.4, random_state=0)
```

```
linear = LinearRegression()
linear.fit(X_train, y_train)
# print("Liner Predicted = ", linear.predict(X_test))
print("linear mean_squared_error = ", mean_squared_error(linear.predict(X_test), y_test))
```

```
linear mean_squared_error =  5783737855.043337
```

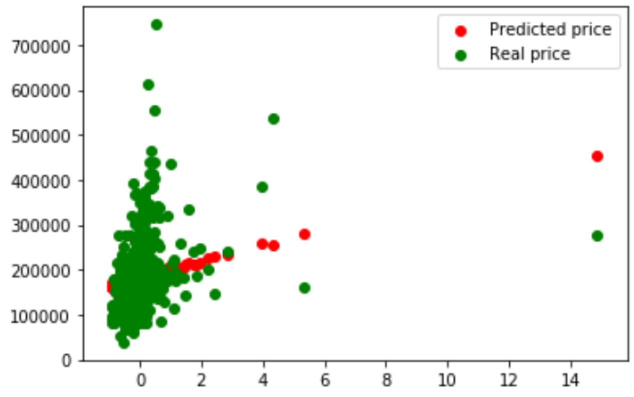
So I use another kind of regression called random forest regression, which is a type of additive model that makes predictions by combining decisions from a sequence of base models, and the mean squared error is also extremely huge, so I thought the model to be not accurate.

```
random_forest_regressor = RandomForestRegressor()
random_forest_regressor.fit(X_train, y_train)
# print("random_forest_regressor Predicted = ", random_forest_regressor.predict(X_test))
print("random_forest_regressor mean_squared_error = ", mean_squared_error(random_forest_regressor.predict(X_test), y_test))
```

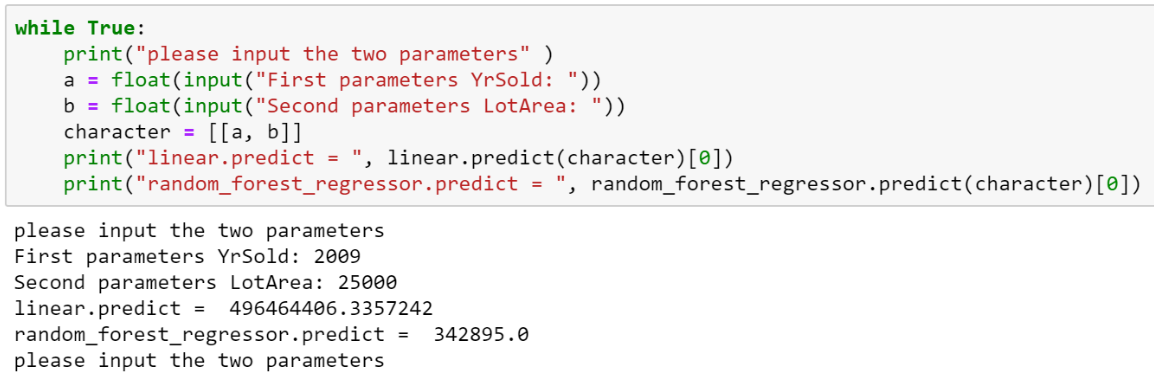
```
random_forest_regressor mean_squared_error =  6179562261.831931
```

Therefore I plot the predict outcome of the liner regression.

```
plt.figure()
plt.scatter(X_test[:, 1], linear.predict(X_test), label = "Predicted price", color = "red")
plt.scatter(X_test[:, 1], y_test, label = "Real price", color = "green")
plt.legend()
plt.show()
```



At the end, I create a simple application that can let people enter information and get an estimate of their house price. There are two parameters needed, the first one is sold year and the second is lot area. After running this program, we can obtain a predictive house price in both two kinds of regression model.



```
while True:
    print("please input the two parameters" )
    a = float(input("First parameters YrSold: "))
    b = float(input("Second parameters LotArea: "))
    character = [[a, b]]
    print("linear.predict = ", linear.predict(character)[0])
    print("random_forest_regressor.predict = ", random_forest_regressor.predict(character)[0])
```

```
please input the two parameters
First parameters YrSold: 2009
Second parameters LotArea: 25000
linear.predict =  496464406.3357242
random_forest_regressor.predict =  342895.0
please input the two parameters
```

In this project, I know more about housing prediction. There could be lots of parameters that can influence the sale price, not only lot area and year sold. I will research more in the future study.