**Final Report**

UK Housing Price Analysis

**Team Member**

Chaoyi Yuan

**NUID**

001347243

**Table of Contents**

1 Introduction

1.1 Background

1.2 Motivation

1.3 Goal

2 Methodology

2.1 ML Algorithms

2.1.1 Multioutput Regression

2.1.2 Multioutput Classification

2.2 Parallel Computing Methods

2.2.1 Multiprocessing, map function for pandas dataframe

2.2.2 Multiprocessing, starmap function for ML algorithm

2.2.3 Multiple Nodes

3 Description of Datasets

3.1 UK Datasets

3.1.1 UK Housing Price

3.1.2 UK GDP

3.1.3 UK Population

3.2 Japan Datasets

3.3 China Datasets

4 Results and Analysis

4.1 Parallel Computing

4.2 Housing Price

4.2.1 Shinjuku House Price

4.2.2 Men Tou Gou House Price

5 Conclusion

6 Reference

1 Introduction

* 1. Background

Nowadays, a house is playing an important role in people’s life. Not only the house is a living place for human beings, but also it is one of the most significant real estate for our financial investment.

Chart, line chart

Description automatically generatedAccording to the development for those years, we can see the housing price is increasing very fast. Based on the picture 1, the housing price in New York City from 1975 to 2021, we can see that the housing price index(HPI) is increasing from about 50 to approximately 150 which is about 3 times than before from 1995 to 2021.

Picture 1. New York City Housing Price from 1975 to 2021

So, we can know that the housing price is becoming a global problem for everyone.

1.2 Motivation

My homeland is China which is a fastest developing country in the world in those 30 years. The income of the people in this land is increasing in those years. According to the picture 2, we can see that the average yearly wages in China is increasing from 37147 RMB to 93383 RMB which is 3 times than before in just 10 years.

Chart, bar chart

Description automatically generated

Picture 2. The Average Yearly Wages in China from 2010 to 2020

However, with this highly income, people are going to get better their life and do some more investments than before. So, it makes people try to invest and spend their money into the real estate field. With the growth of rich people, the price of the house is increasing as well. According to the Picture 3 showed below, we can see that the housing price is increasing from about 80 to 160 in 10 years which is 2 times than before.

Although the house price is increasing with the increasing of people’s income, the problem is in the society, the old people can hold most of the money, not the young people. It makes that the young who is graduating from the school have no enough money to have their own house. And in Chinese culture, every family should have their own house as well. This brings a big pressure for the young people in China.

Chart, line chart

Description automatically generated

Picture 3. The Housing Price in China from 2010 to 2020

Nowadays, some people always ask whether the housing price is too high or not. And does the economic bubble exist in our real estate field? So, based on those questions, I am going to do some research on the real price for the houses.

1.3 Goal

I cannot use the datasets from China, because I cannot find out an entire housing price dataset from 2010 to 2020 in China. After searching the datasets on Kaggle, I decided to use the housing price datasets in UK, because this dataset is from 1995 to 2017 and it saved every house trading in each district in UK.

For my Goal in this project, I am going to arrange the UK housing price datasets and collect the GDP and population data for each district in UK. And then, I am going to train the UK datasets, and then predict the Japanese datasets from 1990 to 2015. Because in that time in Japan, they got an economic disaster which is the Japanese Asset Price Bubble happened in early 1992. I am trying to use the UK housing price training data to predict a reasonable house price at that time.

Except the prediction for Japanese Asset Price Bubble, I am going to predict the house price in Beijing, China. I will pick the smallest district in Beijing which is Men Tou Gou. Men Tou Gou has 345,000 people in 2019. This population is very similar to the districts in UK, and the GDP is similar as well. So, in the end, we can get a prediction for Men Tou Gou housing price. Comparing with the actual house price, we can give an surprised result whether the real estate field in Beijing has economic bubble or not.

2 Methodology

2.1 ML Algorithms

In this project, the target value is 2D array which is not a 1D array as normal, so we cannot use the regular sklearn methods. We can only use the sklearn.Multioutput which is fit for multiple dimension array.

2.1.1 Multioutput Regression

Multioutput regression support can be added to any regressor with MultiOutputRegressor. This strategy consists of fitting one regressor per target. Since each target is represented by exactly one regressor it is possible to gain knowledge about the target by inspecting its corresponding regressor. As MultiOutputRegressor fits one regressor per target it cannot take advantage of correlations between targets.

Python Code:

***from******sklearn.multioutput******import*** *MultiOutputRegressor*

***from******sklearn.ensemble******import*** *GradientBoostingRegressor*

*def regression(X, y, num, X\_predict):*

*return MultiOutputRegressor(GradientBoostingRegressor(random\_state=num)).fit(X, y).predict(X\_predict)*

In this python code, X is training data of eigenvalue which is the population and GDP data in UK. The y is the training data of target value which is the house trading recode 2D list in UK. The num is the random state. And X\_predict is the eigenvalue for prediction such as the population and GDP data in Shinjuku and Men Tou Gou. The return value is the prediction of the target value from eigenvalue of X\_predict via the training data.

In this project, we have two different X\_predict data. One is the population and GDP value in Shinjuku and the other one is the population GDP value in Men Tou Gou, Beijing. The results will be analyzed with the actual house price in Shinjuku and Men Tou Gou.

2.1.2 Multioutput Classification

Multioutput classification support can be added to any classifier with MultiOutputClassifier. This strategy consists of fitting one classifier per target. This allows multiple target variable classifications. The purpose of this class is to extend estimators to be able to estimate a series of target functions (f1,f2,f3…,fn) that are trained on a single X predictor matrix to predict a series of responses (y1,y2,y3…,yn).

Python Code:

***from******sklearn.multioutput******import*** *MultiOutputClassifier*

***from******sklearn.ensemble******import*** *RandomForestClassifier*

*def classification(X, Y, num, X\_predict):*

*n\_samples, n\_features = X.shape*

*n\_outputs = Y.shape[1]*

*n\_classes = 3*

*forest = RandomForestClassifier(random\_state=1)*

*multi\_target\_forest = MultiOutputClassifier(forest, n\_jobs=-1)*

*return multi\_target\_forest.fit(X, Y).predict(X\_predict)*

In this python code, the inputs and output are very similar to the multioutput regression function. We are still going to use the UK population and GDP value to be the eigenvalue of training data, and use the UK house trading records to be the target value of training data. And then we can put Shinjuku and Men Tou Gou data as the X\_predict eigenvalue in the function. In the end, we can get the results of this classification function. Comparing to the previous sklearn method, we can use the results with higher accuracy.

2.2 Parallel Computing Methods

In this project, we have to deal with a huge number of data and use the machine learning algorithm to train the data and then predict the other data. Parallel computing methods play a very import role in this project. I am going to use the parallel computing in three different parts to optimize the program. Except that, I will keep the serial program as well, and set a timer for both serial program and parallel programs. Finally, we can see the competition between the serial compute and the parallel compute in time.

2.2.1 Multiprocessing Pool, map function for pandas dataframe

During figuring out the pandas dataframe data, we have approximately 7.7 million records to deal with. The serial code is showed below:

*start = time.perf\_counter()*

*for i in range(1998, 2018):*

*tmp = str(i)*

*housing\_price.loc[housing\_price['Date of Transfer'].str.startswith(tmp), 'Date of Transfer'] = tmp*

*print("Serial compute, Time elapsed: ", (time.perf\_counter() - start)\*1000)*

Through this program, we can see that it is trying to convert the year value in column ‘Data of Transfer’. The parallel code is showed below:

*# update the time information*

*def update\_year(year):*

*sub\_input = housing\_price.loc[housing\_price['Date of Transfer'] == year]*

*sub\_input['Year'] = year*

*return sub\_input['Year']*

*if \_\_name\_\_ == "\_\_main\_\_":*

*start = time.perf\_counter()*

*pool = Pool(processes=8)*

*# all\_years = [1998, 1999, 2000 … 2017]*

*results = pool.map(update\_year, all\_years)*

*pool.close()*

*pool.join()*

*print("Parallel compute, CPU=8, Time elapsed: ", (time.perf\_counter() - start)\*1000)*

*# concatenate results into a single pd.Series*

*results = pd.concat(results)*

*# join results with original df*

*housing\_price = housing\_price.join(results)*

In this parallel code, the program uses the multiprocessing pool, map function to do the parallel work and uses 8 CPUs in the computing. The function update\_year is trying to pick up each year of data in the pandas dataframe. And then the function can deal with the data for each year and return the result for each year. The multiprocessing pool is trying to parallel different years computing and in the end, join all the results together into the origin pandas dataframe.

The optimizing results will be showed in part 5, Results and Analysis.

2.2.2 Multiprocessing Pool, starmap function for ML algorithm

Next, in this part, I will use the starmap function in multiprocessing pool to parallel compute the machine learning algorithm. The original serial program is already showed in last part. The parallel program is showed below:

*# multioutput regression*

*def regression(X, y, num, X\_predict):*

*return MultiOutputRegressor(GradientBoostingRegressor(random\_state=num)).fit(X, y).predict(X\_predict)*

*if \_\_name\_\_ == "\_\_main\_\_":*

*start = time.perf\_counter()*

*pool = Pool(processes=8)*

*args = [(x\_data, y\_data, num, shinjuku\_x\_data) for num in range(0, 3)]*

*results = pool.starmap(regression, args)*

*pool.close()*

*print("Parallel compute in ml regression, CPU=8, Time elapsed: ", (time.perf\_counter() - start)\*1000)*

In this parallel code, the program uses the multiprocessing pool, starmap function to do the parallel work and uses 8 CPUs in the computing. The function regression is trying to do the regression for the training data with different random seeds. The return value is a 2D array for the prediction.

The optimizing results will be showed in part 5, Results and Analysis.

2.2.3 Multiple Nodes

Using multiple nodes can definitely advance the running time for the program. For just using one node in srun is showed below:

*srun -p express --mem=10Gb --pty /bin/bash*

The memory is set to be 10 Gb, because reading the large csv file has to load a huge amount of memory. Using 8 nodes in srun is showed below:

*srun -N 8 -n 8 -c 8 -p express --mem=10Gb --pty /bin/bash*

To measure the performance of the multiple nodes, we can compare the running time in serial and parallel with different running nodes.

The optimizing results will be showed in part 5, Results and Analysis.

3 Description of Datasets

3.1 UK Datasets

3.1.1 UK Housing Price

Directory: data/price\_paid\_records.csv

Excel Sheet:

Website Link: https://www.kaggle.com/hm-land-registry/uk-housing-prices-paid

The UK housing price dataset is from Kaggle, and the size of this dataset is about 2.4 GB which contains more than 2.2 million house sale records in UK from 1995 to 2017.

Table

Description automatically generatedPicture 4. Reading the UK House Price Dataset

As the picture showed above, we can see the most important columns are “Price”, “Date of Transfer” and “District”. The “Price” column represents the sale price for every house. The “Date of Transfer” represents the sale date. And the “District” means the sale house location.

Next, we need to clean the un-useful data. First, we need to clean the data from 1995 to 1997 which means we should keep the data from 1998 to 2017, because in population dataset, we only have the data from 1998. Second, we need to delete the columns except the price, date and District. Third, not all the districts listed in housing price dataset are in the population and GDP datasets. According to the comparison between house price dataset, GDP dataset and population dataset, I find out that there are 107 districts showing all in three datasets. Now, I can delete the other districts which do not have the GDP and population data. The result is Table

Description automatically generatedshowed below in Picture 5.

Picture 5. Data Cleaning the Dataset

Now, we can see the housing price dataset still have 7,710,382 records of data.

A picture containing text

Description automatically generatedNext, I am going to separate the housing price data based on the different districts and years which means the data with the same district and year will combine together into a list. And then, we can get a 2D array which is like the picture showed below.

Picture 6. House Price Array

Because the number of the house trading record is different in different districts, for example, there are more than 5000 house trading records in Liverpool in 1998. However, there is nothing house trading record in Orkney Islands in 1998.

Next step is to arrange those data and the list. First, we need to make the length of the each list same which means we should delete some lists and records. Then, we need to keep enough lists. So, finally, I decided to set the length of each list to be 886 records, which left 2000 lists is the 2D array. And this 2D array will be the x\_data in the end. The length of x\_data is 2000 and the length of x\_data[i] is 886.

3.1.2 UK GDP

Directory: data/regionalgrossdomesticproductgdpcityregions.xlsx

Excel Sheet: Table 5

Website Link: https://www.ons.gov.uk/economy/grossdomesticproductgdp/datasets/regionalgrossdomesticproductcityregions

A picture containing table

Description automatically generatedThis dataset shows the GDP data for each district in UK. We can see the picture showed below. Every district has its own GDP data from 1998 to 2018.

Picture 7. Reading GDP Dataset

Graphical user interface, application, table, Excel

Description automatically generatedNow, we can do the data cleaning for this GDP dataset. First, we do not need the 2018 data, because we do not have the 2018 house trade data in previous dataset. Second, we do not need the data with CR in Area type, because CR is the recognition of district in UK, we only need the data with LA in Area type. Third, we should uppercase all the area name in the dataset. The result is showed below.

Picture 8. Data Cleaning GDP Dataset

3.1.3 UK Population

Directory: data/regionalgrossdomesticproductgdpcityregions.xlsx

Excel Sheet: Table 6

Website Link: https://www.ons.gov.uk/economy/grossdomesticproductgdp/datasets/regionalgrossdomesticproductcityregions

This dataset shows the population data for each district in UK. We can see the picture showed below. Every district has its own population data from 1998 to 2018.

A picture containing text

Description automatically generated

Picture 9. Reading Population Dataset

Graphical user interface, application, table

Description automatically generatedNow, we can do the data cleaning for this population dataset. First, we do not need the 2018 data, because we do not have the 2018 house trade data in previous dataset. Second, we do not need the data with CR in Area type, because CR is the recognition of district in UK, we only need the data with LA in Area type. Third, we should uppercase all the area name in the dataset. The result is showed below.

Picture 10. Data Cleaning Population Dataset

Then, we can use the population and GDP data to do the x\_data for machine learning algorithm. Comparing with the y\_data we have done before, we should redo a x\_data corresponding to the y\_data. The x\_data will be a 2D array as well. In each list, the first element is GDP data, and the second element is population data. The length of the x\_data is 2000 which is same as the length in y\_data, and the length of the x\_data[i] is 2 for each list.

3.2 Japan Datasets

Directory: data/prediction\_data.xlsx

Excel Sheet: Shinjuku

House Price Website Link:

https://japanpropertycentral.com/2013/10/japans-apartment-market-over-the-last-40-years/#:~:text=In%201991%2C%20apartment%20prices%20peaked,in%201990%20at%2061%2C230%2C000%20Yen.

https://resources.realestate.co.jp/buy/tokyo-residential-real-estate-up-or-down-in-2015/#:~:text=In%20the%20first%20three%20quarters,35%2C310%2C000yen%20(about%20%24293%2C000).

GDP Website Link: https://www.hilife.or.jp/datafile2017/tatsuzawa\_2017\_06.pdf

Population Website Link: http://demographia.com/db-tokyo-ward.htm

Graphical user interface, text, application, table, Excel

Description automatically generatedFor picking the house price data in Japan, in the beginning, I planned to catch the dataset in Tokyo. However, as we know, Tokyo is one of the largest cities around the world. Tokyo has about 30 million people now which means the Tokyo dataset cannot fit for the training data in UK, because each district in UK has only 100 kilo to 500 kilo people. In the end, I tried to catch the dataset for the district in Tokyo which is Shinjuku. Shinjuku is a very famous district in Tokyo, and its economy can show the growth and progress in Tokyo. As a part of district in Tokyo, I believe Shinjuku can represent Tokyo. The Shinjuku GDP, population and house price data are showed below

Picture 11. Shinjuku Data

As we know, the Japanese asset price bubble happened in early 1992. The peak of the house price appeared in 1990 to 1991 and then, the economic bubble burst, the house price was staying at a low price for almost 30 years.

For simulating this economic bubble, I pick up the data from 1990 to 2015. And I do convert the Japanese Yen to English Pound based on the exchange rate in each year.

3.3 China Datasets

Directory: data/prediction\_data.xlsx

Excel Sheet: Men Tou Gou

House Price Website Link: https://www.anjuke.com/fangjia/beijing2011/mentougou/

GDP Website Link: https://www.ceicdata.com/zh-hans/china/gross-domestic-product-municipality-district/cn-gdp-beijing-mentougou

Population Website Link: https://www.ceicdata.com/zh-hans/china/population-municipality-district/cn-population-beijing-mentougou

The goal for this project is to use the training data in UK to predict the Japanese house price during the economic bubble. And then, go back to my homeland to see whether China has a house price bubble in the market. As we know, Beijing is the capital of China and Beijing has the one of the highest house prices in China. If we can get a result from Beijing, we can see the bubble is exist or not.

Beijing is a very large city in the world as well. We can just pick a district in Beijing, Men Tou Gou(门头沟) district is a best choice. Men Tou Gou is a suburban district in Beijing and it has the fewest population in Beijing which is about 330k.

Graphical user interface, application, table, Excel

Description automatically generated

Picture 12. Men Tou Gou Data

The Picture showed above is the GDP, population and actual house price in Men Tou Gou from 2006 to 2019. All the Chinese Yuan has been transferred to UK pound based on the exchange rate in each year.

4 Results and Analysis

In this part, we will show the results and analyze the results. We will talk about it through two directions. One is the optimizing performance of the parallel computing in the program. The other one is to summarize the housing price in Shinjuku and Men Tou Gou, and then analyze the results for the prediction.

4.1 Parallel Computing

Setting the number of nodes to 1, and then run the python file named “parallel\_compute.py” on cluster. After 15 minutes, we can an output in console, and we can see the picture below.

Text

Description automatically generated

Picture 13. Console Output with One Node

And then setting the number of nodes to 8, and then run the python file, “parallel\_compute.py” on cluster. We can get an output in console, the picture showed below.

Text

Description automatically generated

Picture 14. Console Output with Eight Nodes

After rearranging the outputs above, we can get a table here:

Graphical user interface, application, table, Excel

Description automatically generated

Picture 15. Running Time in Serial and Parallel

Comparing vertically, during dealing with the pandas dataframe, we can see the serial computing will take much more time than parallel computing program. With only one node running, the running time for serial computing program is 6.33 times than the running time for the parallel computing program. During figuring out the machine learning regression, the running time for serial computing program is 1.5 times than the running time for the parallel computing program

Comparing horizontally, during figuring out the pandas dataframe, with 8 nodes running, the running time for serial computing program is 28.11 times than the running time for the parallel computing program. And during dealing with the machine learning regression, with 8 nodes, the running time for serial computing program is 2.94 times than the running time for the parallel computing program

This shows two obvious points. One is that during dealing with the pandas dataframe, the parallel computing program does optimize the running time. The second is that with multiple nodes working, the parallel commuting program will be advanced a lot.

4.2 Housing Price

4.2.1 Shinjuku House Price

Chart, histogram

Description automatically generatedAs we know, the economic bubble appeared in early 1992, the house price was falling down like a cliff in Tokyo. We can see the picture.

Picture 16. Tokyo Average House Price from 1973 to 2012 in Japanese Yen

The unit is Japanese Yen. We can see from 1973 to 2012, the graph is like a mountain. The peak is in 1990. However, if we consider about the exchange rate between Japanese Yen and English Pounds, this graph will be totally different.

Graphical user interface, chart, line chart

Description automatically generatedPicture 17. Japanese Yen to English Pound from 1990 to 2020

According to the picture, we can see that although the house price in 1990 is the peak in Japanese Yen, combining with exchange rate with Pound, the actual house price is lower than the house price in 1995. Connecting to the next part, I am going to show the results of the prediction and see the relationship.

After doing the machine learning regression, we get a 2D array of house price records. Then, we can use those data to calculate the average house price in each year. In the end, we can use those data to compare with the actual average house price in each year and plot it in python. The picture is showed below. The blue line represents the actual house price in Shinjuku in English Pounds, and the orange line represents the predicational house price in Shinjuku in English Pounds. We can see the trend of two lines are very similar. It means the even in the economic bubble in Japan, this machine learning model is still working very well. The Japanese housing price follows the training data in UK very well.

Chart, line chart

Description automatically generatedPicture 18. Actual and Predicational House Price in Shinjuku

Chart, line chart

Description automatically generated4.2.2 Men Tou Gou House Price

Picture 19. Actual and Predicational House Price in Men Tou Gou

After prediction to the population and GDP data in Men Tou Gou, we can get the results of the prediction. And then calculate the average house price for each year and plot it.

According to the picture, we can see the prediction values are far lower than the actual house price. Even in the 2012, the closest house price year, the actual house price is more than 3 times than the prediction value.

5 Conclusion

This project has achieved the goals we made before. First is to measure the optimization performance of parallel computing. The second is to prove whether there is an economic bubble in Chinese real estate field or not.

In conclusion, the parallel computing including the multiprocessing and multiple nodes do advance the running time for the programs. And based on the Tokyo house price analysis, we can know that the training model is correct, and it can fit for the Japanese house price from 1990 to 2015. Through predicting the house price in Men Tou Gou district, we can know that there must be an economic bubble in real estate field in Beijing. With the low GDP and large number of people, the house price should not be this kind of high.

6 Reference

(macrotrends, 2020)

(scikit learn, n.d.)

(Office for National Statistics, 2020; CEIC, 2020)

(CEIC, 2020)

(CEIC, 2020)

(Demogr Aphia, 2006)

(東京の「経済力」, 2020)