Irene(Jiaying) Xu, UAL Creative Computing Institute

23/24 Introduction to Data Science I
In-Class Assignment Portfolio - Exercise 2: Predictive Analysis-Regression

0 Introduction

**Related Course Material: Week 5 Regression*

In Week 5, we looked at how to find the relationships between variables and build models for prediction (numerical values). A new dataset was adapted in this exercise to further practice and explore the methods of building predictive models.

1 Dataset Prep & Overview

The dataset used for this exercise was found on Kaggle - "*Housing in London*" created by Justinas Cirtautas. It includes UK housing price information from 1995 to 2019, along with other relevant data such as salary and population size. The dataset was briefly used in Exercise II to generate some animated plots, here using the same dataset, the goal is to delve deeper into the relationships between the features (variables) provided in the dataset.

1) Data Pre-processing

   The original dataset has split into two csv files based on the variable collection frequency (monthly and yearly). The steps I have taken to prepare a dataset for conducting regression tasks include:

   a. Keep the rows with `borough_flag` = 1 (others are not labelled by boroughs)
   b. Convert datetime to the year format (e.g., 1999-12-1 to 1999).
   c. Calculate the avg. annual housing price based on the `date (monthly)` and `avg. housing price` in housing_in_london_monthly_variables.csv. Create a new df containing only the year, the area, and avg. housing price.
   d. Merge the new sf with housing_in_london_yearly_variables.csv based on the same date and area, anything with no match also gets dropped.
   e. Inspect the merged dataset and drop columns with missing data.

   The merged dataset is shown below:

	code	date	area	average_price	median_salary	mean_salary	population_size
0	E09000001	1999	city of london	171300.08333	33020.00000	48922.00000	6581.00000
1	E09000002	1999	barking and dagenham	65320.83333	21480.00000	23620.00000	162444.00000
2	E09000003	1999	barnet	136004.41667	19568.00000	23128.00000	313469.00000
3	E09000004	1999	bexley	86777.66667	18621.00000	21386.00000	217458.00000
4	E09000005	1999	brent	112157.41667	18532.00000	20911.00000	260317.00000
...	...	...	...	...	...	...	...
655	E09000029	2018	sutton	379262.58333	28853.00000	32442.00000	204525.00000
656	E09000030	2018	tower hamlets	446500.41667	49237.00000	69806.00000	317705.00000
657	E09000031	2018	waltham forest	440859.41667	30298.00000	32875.00000	276700.00000
658	E09000032	2018	wandsworth	596649.16667	34501.00000	45317.00000	326474.00000
659	E09000033	2018	westminster	1020025.50000	43015.00000	63792.00000	255324.00000

2) Dataset Overview

The merged dataset contains 645 rows and 7 columns, covering 33 unique areas in London, which matches the total number of London boroughs. It provides information on avg. housing price, area median salary, area mean salary, and area population size from 1999 to 2018, each year has about approximately 30 entries, with a minimum of 29 and a maximum of 33, indicating that some areas may be missing from this dataset, but the majority of them are present.

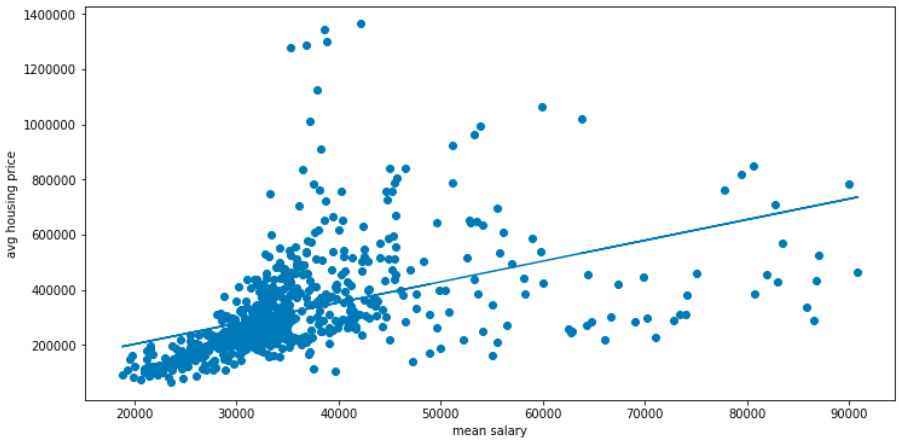
```
array(['city of london', 'barking and dagenham', 'barnet', 'bexley',
       'brent', 'bromley', 'camden', 'croydon', 'ealing', 'enfield',
       'greenwich', 'hackney', 'hammersmith and fulham', 'haringey',
       'harrow', 'havering', 'hillingdon', 'hounslow', 'islington',
       'kensington and chelsea', 'kingston upon thames', 'lambeth',
       'lewisham', 'merton', 'newham', 'redbridge',
       'richmond upon thames', 'southwark', 'sutton', 'tower hamlets',
       'waltham forest', 'wandsworth', 'westminster'], dtype=object)

array([1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009,
       2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018], dtype=int32)
```

2 Correlation

To begin exploring associations between variables, I have started with examining various combinations of 2 variables, with one always being the target value - average housing price.

a) `Mean_Salary` vs. `Average_Price` - a moderate positive correlation



Based on the plot, we could say there is probably a positive correlation between the mean salary and average housing price in the area. Areas with higher mean salaries also tend to have a higher average housing price.

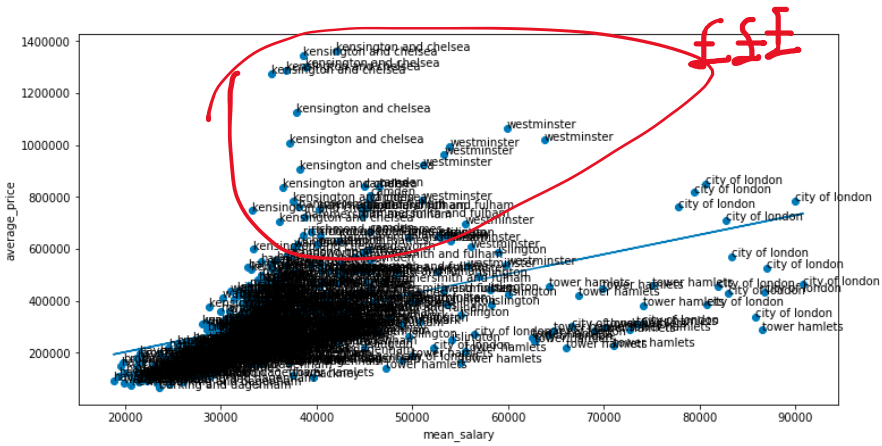
A correlation coefficient (r) of 0.46875 indicates a moderate positive linear relationship between the mean salary and avg. housing price.

The model of this relationship between the mean salary and average housing price can be represented by a linear equation: Avg. Housing Price = 7.5272665 * Mean Salary + 52089.70510250382 + error.

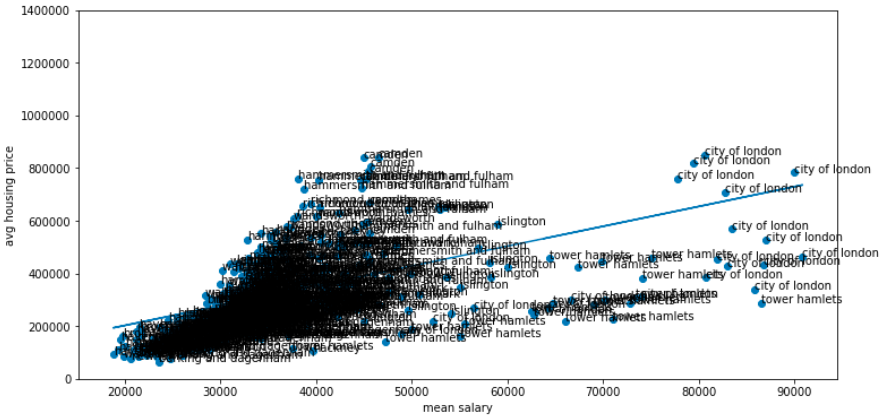
The p-value is also extremely small here, implying strong evidence against the null hypothesis.

It is also obvious that the dataset has some extreme values for housing prices, particularly around a mean salary of 40,000 - the housing price in this mean salary range appears to be much higher compared to the rest of the data.

By annotating the data points, we can see that those extreme values belong to either `kensington and chelsea` or `westminster` areas, which makes sense as these areas are known for their high property values in London.



In this case, will removing those outliers affect the relationship between mean salary and average housing price? (improve the model??)



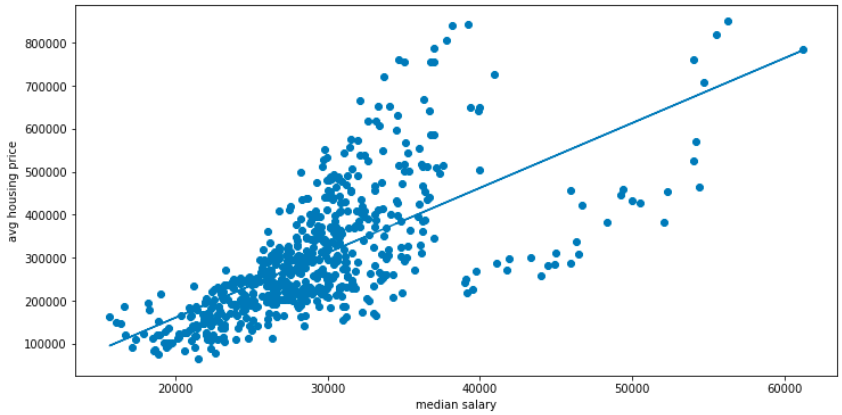
After removing data of 'kensington and chelsea' and 'westminster', the correlation coefficient (r) increased from 0.468 to 0.53578. It is not a substantial change but could indicate a slightly stronger positive linear relationship between mean salary and average housing price compared to what's found in the original dataset.

Nevertheless, this higher correlation coefficient DOES NOT mean the model is better, those extreme data points are real data that could represent real housing price patterns or trends (e.g., probably relates more to the social or cultural background of the area? rather than salary and population size, but such factors are not included in the dataset) - removing real data simply not help but hurt the model's robustness in prediction. It would be better to explore alternative methods (e.g., robust linear regression?) that de-emphasise the outliers instead of removing them completely. In general, I have learnt that outliers should not be removed unless 1. They are confirmed errors that cannot be

fixed, or 2. They are not part of the population I wish to find patterns/trends or gain insight.
**reference posts: StackExchange - Will removing outliers improve my predictive model?; Statisticsbyjim - Guidelines for Removing and Handling Outliers in Data*

Looking at all the data points, I would say that a linear correlation here isn't going to be the best option for modelling the relationship between the mean salary and avg. housing price (it might be okay if we exclude the Kensington, Chelsea, and Westminster Areas). Other modelling techniques such as regression should be explored to see if they can better capture the underlying relationship.

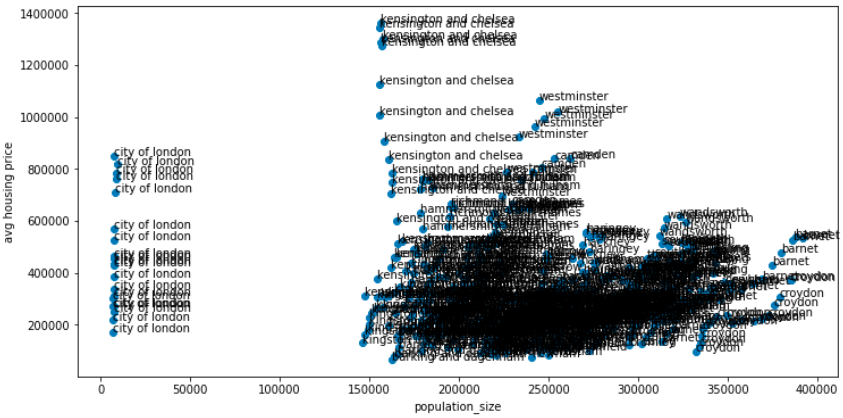
b) `Median_Salary` vs. `Average_Price` - a moderate to strong positive correlation



The above plot shows a positive correlation between median salary and average housing price. A correlation coefficient (r) of 0.57173 without data from Kensington, Chelsea, and Westminster) indicates a (slightly) stronger positive linear relationship between the median salary and average housing price compared to the relationship between the mean salary and average housing price.

The model can be represented by Avg. Housing Price = 16.86669 * Median Salary - 170146.76204597624 + error

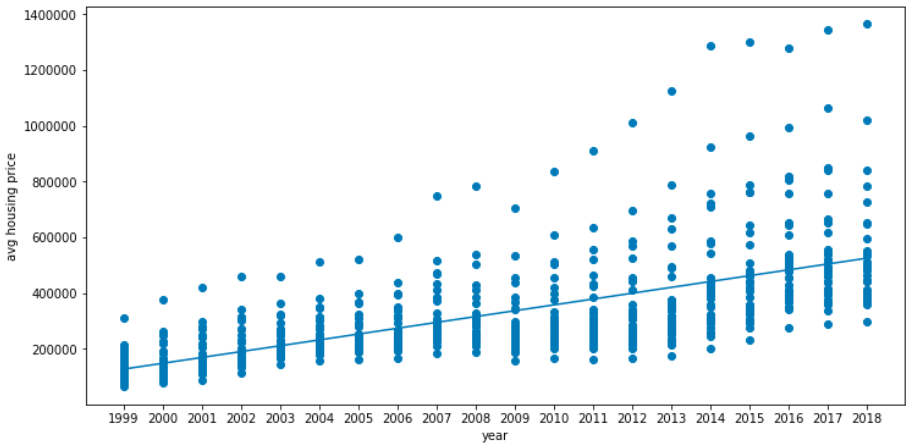
c) `Population_size` vs. `Average_Price` - a slight negative linear relationship / no relationship - NOT a good predictor



Then, for the population size and average housing price, a correlation coefficient (r) of -0.13548447470395442 indicates that the relationship overall is very weak and might not be useful for predictive modelling. Nevertheless, there are still some interesting findings that can be drawn from the plot:

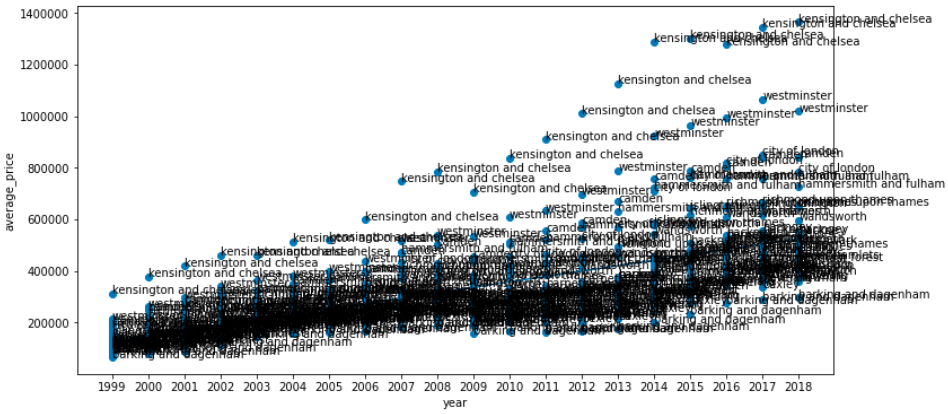
- The City of London area, despite having the lowest population size, exhibits a relatively high average housing price compared to other areas (makes sense as it's the city centre).
- Kensington, Chelsea, and Westminster have a moderate population size but significantly higher average housing prices.
- Areas such as Barnet and Croydon have the highest population sizes but relatively lower average housing prices.

d) `Date` vs. `Average_Price` - a moderate to strong positive correlation



The above clearly illustrates a positive correlation between the date (year) and average housing price in London - the avg. housing prices are getting higher and higher each year.

A correlation coefficient (r) of 0.63341 indicates a moderate to strong positive linear relationship between the date (year) and average housing price.



By annotating the graph, we can also see that each year, the areas such as `kensington and chelsea` and `westminster` constantly have the highest avg. housing price (much higher than other areas), while `barking and dagenham` seems to have the lowest avg.

housing price each year (I looked it up on Google Maps and found that it's far away from the City of London, next to London City Airport).

While the linear functions could provide a (very) rough prediction of average housing price based on either area median salary or time, it's important to note that a single independent variable is often insufficient to capture the entire relationship (especially for housing prices that are affected by multiple factors). Therefore, it is also necessary to consider multiple variables in our analysis.

3 Regression

1) Linear Regression

Taking `Population_Size` and `Median_Salary` vs. `Average_Price` as an example, a multiple variables linear regression model can be created by using *linear_model.LinearRegression()* in scikit-learn. (part of the code of the 3D plot is initially adapted from ChatGPT with the prompt - how to plot regression surface, with modification made according to the dataset).

```
x = df[['median_salary','population_size']]
y = df['average_price']
regr = linear_model.LinearRegression()
regr.fit(x, y)
print(regr.coef_, regr.intercept_, regr.score(x, y))
```

✓ 0.0s

```
[16.93660629  0.02596908] -178458.92072258325 0.326961907471254
```

Based on the model, if we have an area that has a median salary of £40,000 and a population size of 200,000, we could expect an average housing price of £ 504,199. (So that an individual would need to work approximately 13 years to afford a house without factoring in any living expenses...).

The evaluation metrics suggest that the model has a relatively low r squared value and high errors. By looking at those stats alone, it seems that the model has a limited

predictive capability and might not be suitable for prediction. Other combinations of variables could be tested to see if we can find a better model.

```
r squared: 0.326961907471254
mean absolute error: 97425.31480318085
mean_squared_error: 24206956360.626804
root mean squared error: 155585.84884438175
```

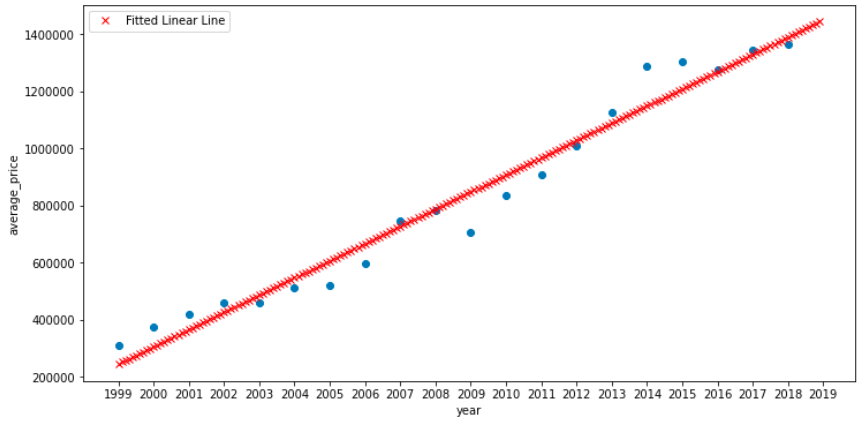
Besides, though have successfully generated a 3D plot of the linear regression model, it is actually really hard to interpret the graph compared to 2D plots. The data points all seem to be piled up and some points are covered by others, making it difficult to tell the relationship between all variables/points – if we have to show 3D plots in future, it might be helpful to make them interactive or aminated so that the plot can be rotated to show all the points clearly.

2) Polynomial Regression

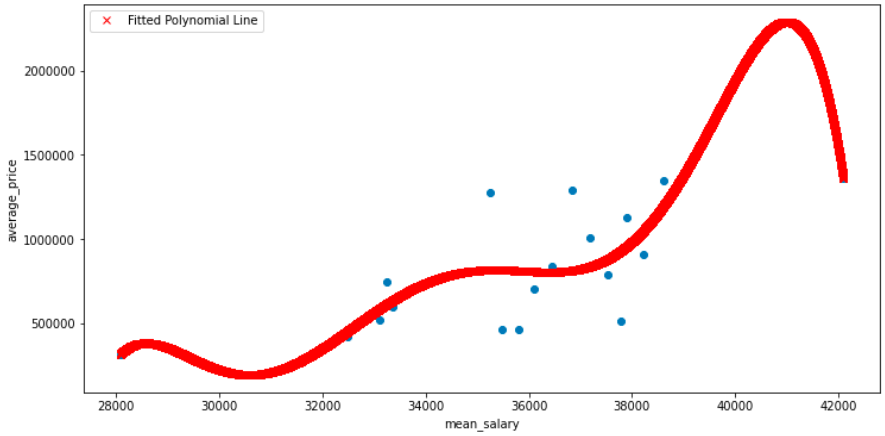
To further explore the relationship that cannot be captured by a linear line – it is also necessary to look at polynomial regression. Here I only took the data from the Kensington and Chelsea area as an example:

In the case of `date` & `avg. housing price` - a linear model already has a very good fit to the data (r2 = 0.9648334724524961).

The graph shows that the average housing price in Kensington and Chelsea will likely continue to rise, but that is not actually the case. According to data from HM Land Registry Open Data that covers the period from 1999 to 2024, the average housing price has been fluctuating rather than showing a consistent upward trend since 2015. While the graph validates the growth observed between 1999 and 2019, it fails to accurately predict future prices. This highlights the limitation of relying solely on past performance for making predictions. Also, housing prices are influenced by a multitude of factors (e.g., cultural and historical backgrounds, economic conditions, government policies, demographics, etc.), as we only have a limited set of factors and not enough data here, it is hard to capture the complexity of these relationships and trends. We need a more comprehensive dataset that includes various relevant variables and also continuously updates the model (if possible) to make more robust predictions.



Nevertheless, in another case of `mean salary` & `avg. housing price` - when I tried to fit the data within a polynomial function, the best r2 I could get is to set the number of degrees of 5 or 6, which gives a r2 around 0.59; I can get a higher r2 by increasing the number of degrees to a very large number, but then the fitted line doesn't make more sense, which I would assume it has limited capabilities to make meaningful predictions as the model is clearly overfitting.



I could probably apply regularisation techniques or gather more data, but overall, except for having a dataset as comprehensive as possible, finding the balance between model complexity, fit, and generalisability is also important, and it is challenging and highly dependent on the specific problem we'd like to solve. Data science is not like math which often has a definitive function or 'best' solution, solutions or models in data science are often iterative, and the best solution might not exist at all, but there is usually room for improvement, by using new data, new techniques or methods. And sometimes, data science is probably not about giving answers but instead, asking better questions, e.g., is there missing data and why, what biases, what is the purpose of the analysis (is it a pure exploratory data analysis, or is there a specific story that the business wants the data to tell, etc.).

4 Code, Supporting Document, and LLM Disclaimer

All datasets, code, Jupyter Notebooks, and GIF files are available to access via the GitHub repository: https://git.arts.ac.uk/23001934/ds-portfolio

ChatGPT 3.5 has been used in this exercise for some proofreading and for searching certain Python codes which have not been taught in class (especially for the part of plotting regression surface in 3d linear regression).

5 References and External Resources

External Resources (Blogs, Posts, etc.):

[1] https://www.geeksforgeeks.org/drop-rows-from-the-dataframe-based-on-certain-condition-applied-on-a-column/

[2] https://stackoverflow.com/questions/893657/how-do-i-calculate-r-squared-using-python-and-numpy

[3] https://statisticsbyjim.com/basics/remove-outliers/

[4] https://stats.stackexchange.com/questions/298551/will-removing-outliers-improve-my-predictive-model#:~:text=into%20an%20answer%3A-,If%20the%20extreme%20values%20are%20not%20errors%20in%20the%20data,retain%20them%20in%20your%20dataset.

[5] https://www.kaggle.com/code/zhehaoz/stat-504-3d-plot-regression-surface-python

[6] https://stackoverflow.com/questions/1985856/how-to-make-a-3d-scatter-plot

[7] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

[8] https://jakevdp.github.io/PythonDataScienceHandbook/04.12-three-dimensional-plotting.html

[9] https://en.wikipedia.org/wiki/London_boroughs

**Detailed annotations were included in the Jupyter Notebook along with the code to indicate where ChatGPT and external resources, such as StackOverflow posts and other Python study materials were used.*