Analyzing and Predicting London Home Prices from Free, Public Data

A Springboard Capstone project by David Bender

March 25, 2020

The Purpose:

Real estate investment firms have to incorporate many factors when deciding where and how to invest. To simulate this challenge, in particular for residential real estate, this analysis will look at London residential data to identify patterns, generate insights, and develop predictive models for price.

- **Data:**

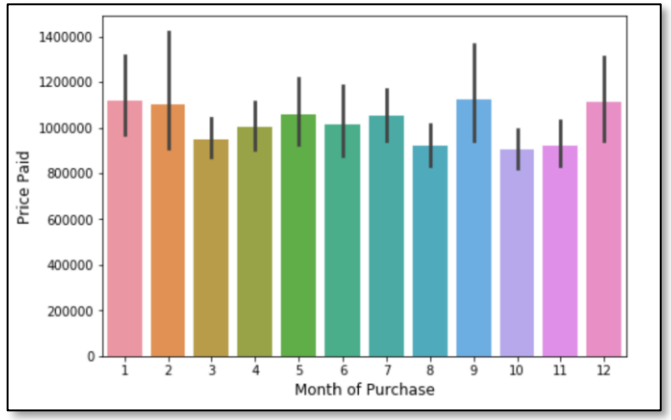
https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads

- **Code notebook:**

https://github.com/dudikbender/springboard/blob/master/Springboard%20Capstone%201%20-%20London%20Home%20Prices.ipynb

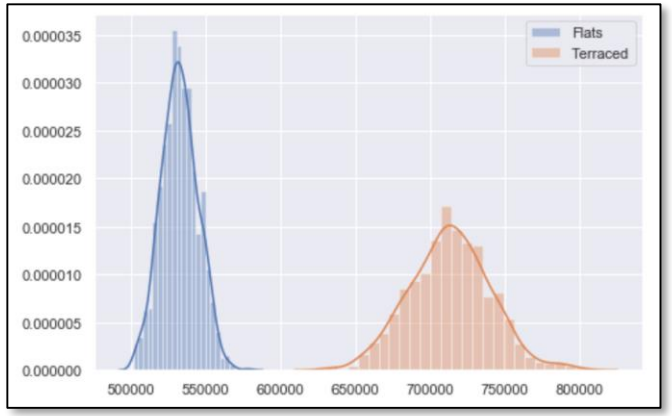
The analysis process

1 Examining the Data



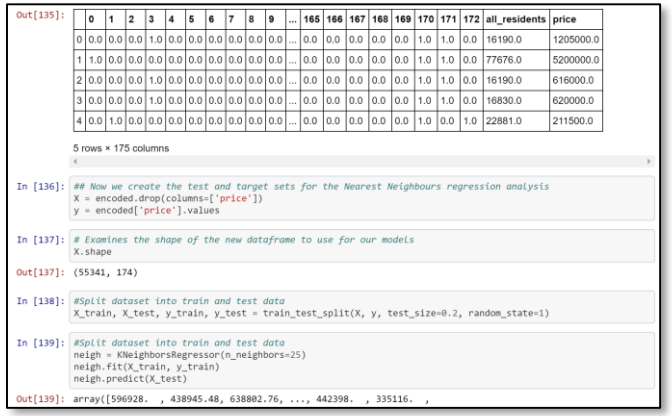
First, we excerpted London data from a national UK dataset of prices paid for residential properties. We cleaned, re-labelled, and examined the data in multiple ways.

2 Statistical Analysis & Mapping



Next, we looked more deeply at the data – including visualizations, inferential statistical analysis, and mapping the locations using the *folium* package.

3 Predictive Modelling



Out[135]:

	0	1	2	3	4	5	6	7	8	9	...	165	166	167	168	169	170	171	172	all_residents	price
0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	16190.0	1205000.0
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	77676.0	5200000.0
2	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	16190.0	616000.0
3	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	16830.0	620000.0
4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	22881.0	211500.0

5 rows × 175 columns

```
In [136]: ## Now we create the test and target sets for the Nearest Neighbours regression analysis
          X = encoded.drop(columns=['price'])
          y = encoded['price'].values

In [137]: # Examines the shape of the new dataframe to use for our models
          X.shape

Out[137]: (55341, 174)

In [138]: #Split dataset into train and test data
          X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

In [139]: #Split dataset into train and test data
          neigh = KNeighborsRegressor(n_neighbors=25)
          neigh.fit(X_train, y_train)
          neigh.predict(X_test)

Out[139]: array([596928.  , 438945.48, 638802.76, ..., 442398.  , 335116.  ,
```

Lastly, we performed machine learning regression analysis utilizing scikit-learn – including KNeighborsRegressor and DecisionTreeRegressor.

Examining the data

Steps taken to prepare and examine the dataset:

- ‘Chunking’ when importing because the original file was quite large (~22M lines)
- Renaming columns for easier readability
  - E.g., home types such as ‘Flat’ or ‘Terraced’ were indicated with values ‘F’ and ‘T’
- Adding new columns, including month and year of purchase, and postcode district (the first 4 characters of a postcode in the UK)
- Adding features related to population density from the UK Datastore, including total population and school-age children population by postcode district
- Creating charts to visualize the distributions of prices by varying features – such as month purchased, price & population density by property type (fig. 1), and histogram of prices by property type (fig. 2)
- Also, we filtered the data using pandas for some analysis (particular the mapping) to only include properties in selected Islington postcode districts

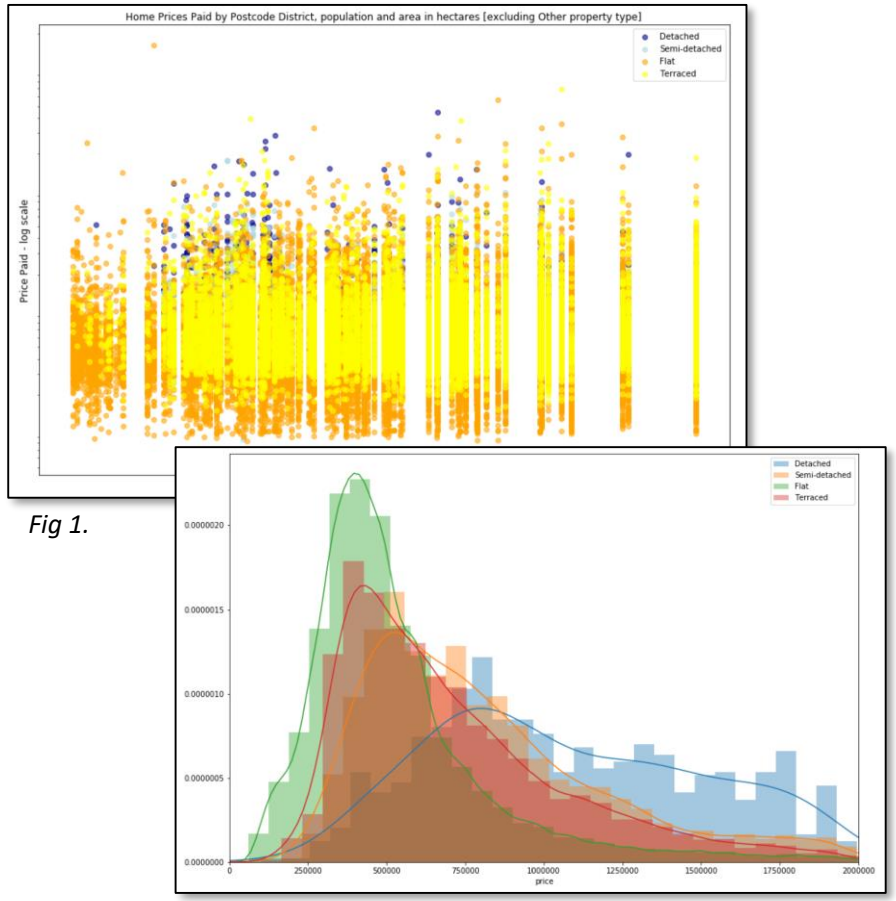


Fig 1.

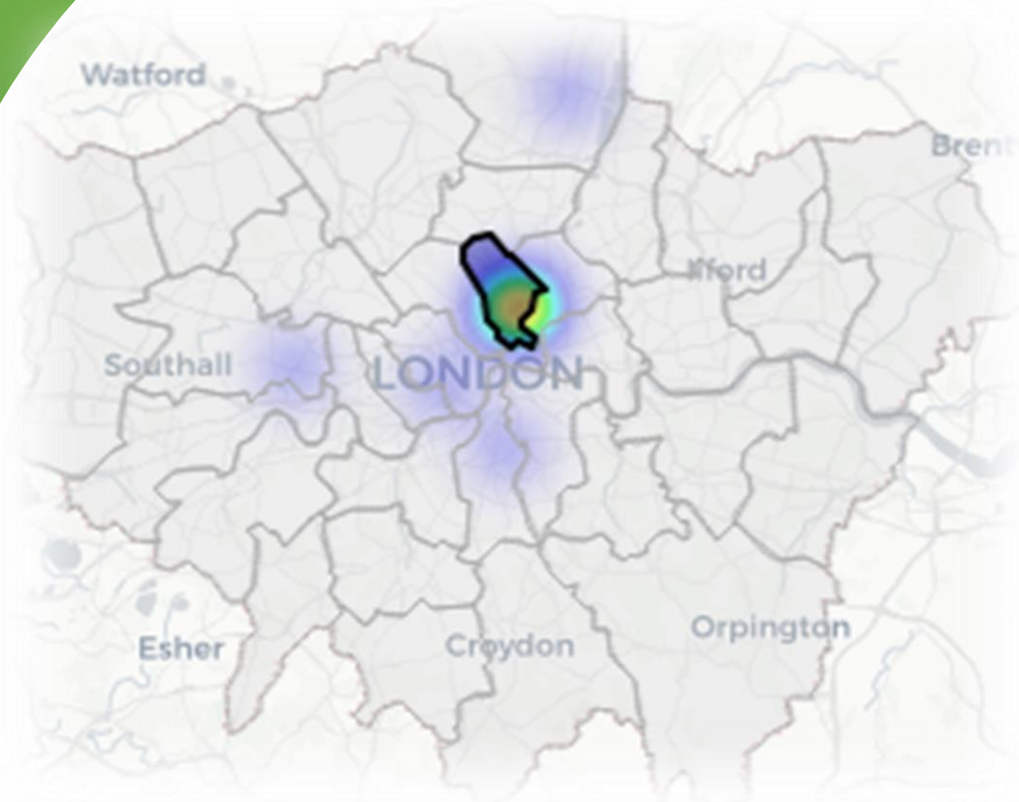
Fig 2.

Statistical Analysis and Mapping

- **Visualizing and analysing the data statistically and geographically:**

- **Approaches;**
  - Bootstrapped inference for comparison of ‘Flat’ and ‘Terraced’ property types
  - Geocoding the address information to extract latitude and longitude co-ordinates
  - Using the *folium* mapping package in Python to create heatmaps of property locations, overlaid onto additional map layers of London borough boundaries (see image to right)
  - Adding interactivity to the map to, for example, allow the user to hover over boroughs for tooltip information or toggle the visibility of layers

- **Some conclusions;**
  - There are statistical differences between prices for ‘Flat’ and ‘Terraced’ properties – interesting as ‘Terraced’ is not a property type used in North America!
  - More data is needed to correctly identify the co-ordinates for some properties – as can be seen the map some properties were outside of the Islington postcodes that we had filtered for. But the resulting maps are quite nice and the location data could be used for even more advanced feature creation using location analysis packages such as geopandas.

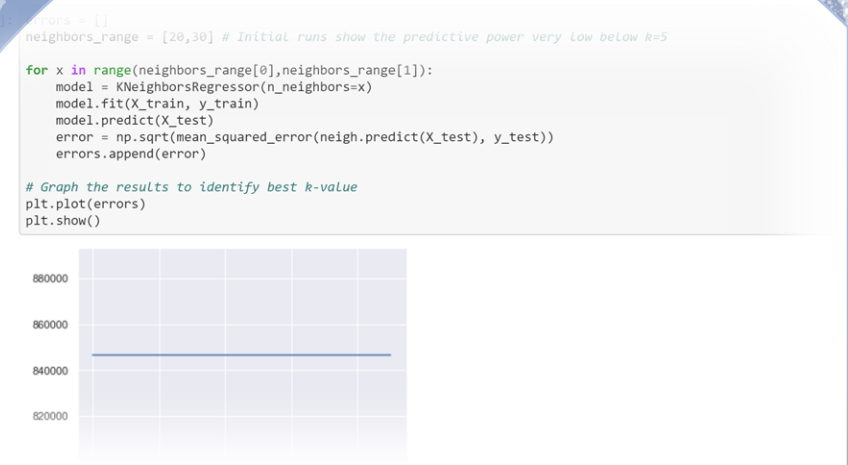


Predictive Modelling

Applied machine learning models using scikit-learn;

- **Approaches;**
  - Preprocessed the categorical data using scikit-learn's OneHotEncoder method (after eliminating the property type 'Other' because its value were not reflective of typical residential properties)
  - Split the data into training and test sets using train_test_split
  - KNeighborsRegressor, with a for loop to determine optimal k-value
  - DecisionTreeRegressor
- **Conclusions;**
  - We used mean squared error to measure accuracy
  - The models did not provide good accuracy, likely due to the features of the dataset not being highly predictive; however, a Decision Tree approach did have a lower error than Kneighbors

Next Steps: develop the dataset to include more predictive features



CONCLUSIONS

The features in the data have limited predictive power. There are insights to draw but more features will need to be included to make any predictive model useful to generate strong estimates of price.

POSSIBLE NEXT STEPS

- Identify other datasets with additional features with higher predictive power, such as number of rooms or London tube stations
- Generate new features using location analysis, such as proximity to transit or points of interest, using packages such as geopandas