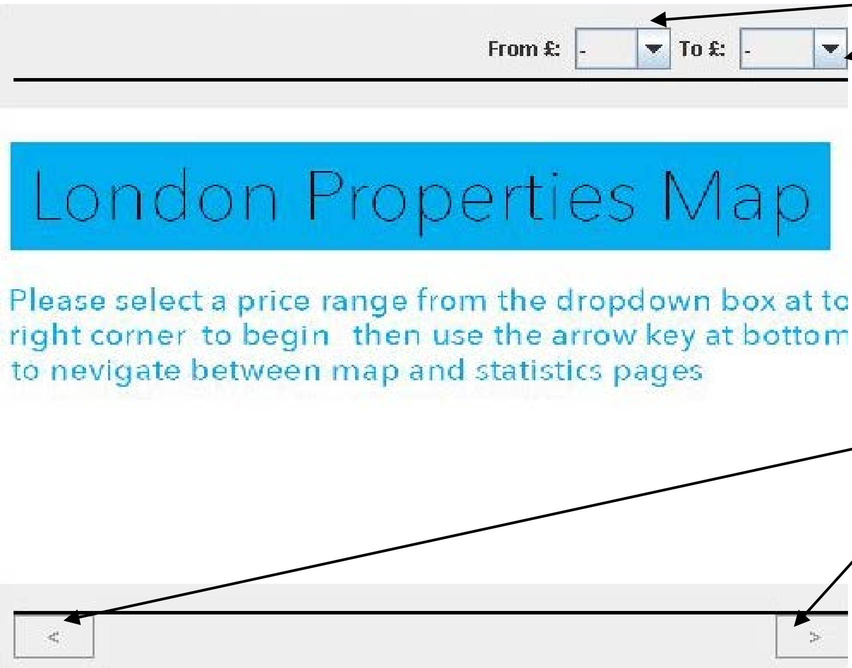
TEAM MERCURIALS - FU ZHANG -1704589 - K1763499; A K M NAHARUL HAYAT -1750583 -K1764014 ;
NOYAN RAQUIB -1786511 -K1764201; MICHAEL OWEN JONES –1722111 - K1763592;

GUI Description

To run the program in bluej, create an object of the start frame class in bluej

Also, additional details can be found in the README.MD file

Welcome Screen:



From £: - To £: -

London Properties Map

Please select a price range from the dropdown box at to right corner to begin then use the arrow key at bottom to nevigate between map and statistics pages

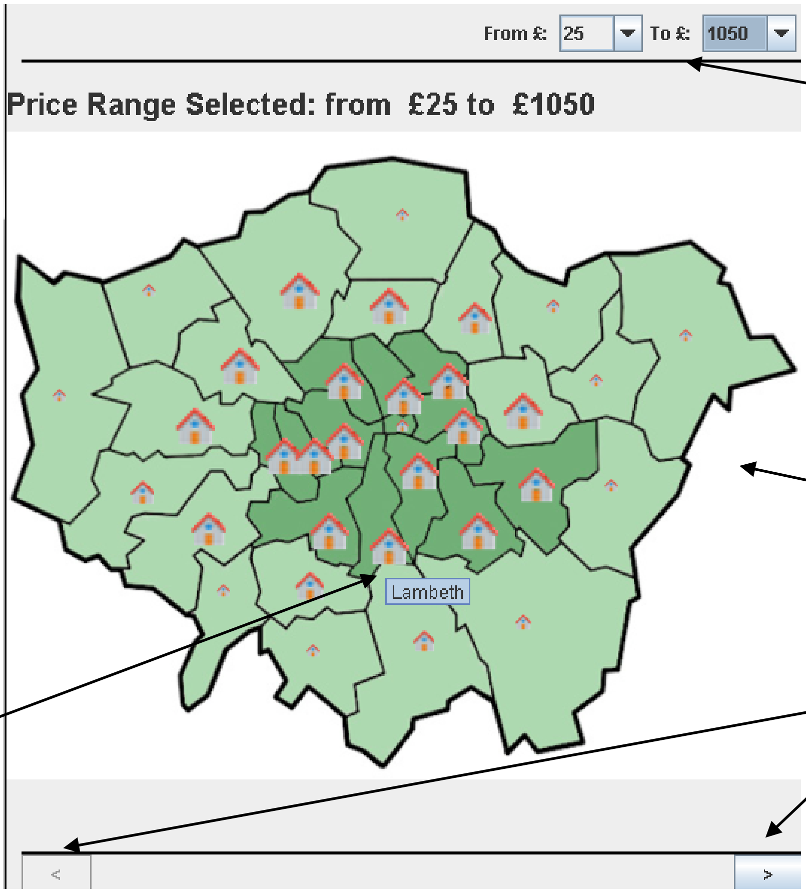
The user can select the price range to search using these 2 self-explanatory comboboxes. Once a valid range is selected, they are taken to the map panel illustrated below

The two panel scroll buttons are initially disabled. They are only enabled after the user selects a valid ‘from’ and ‘to’ value from the combobox above

Map Panel

Each marker is clickable. If you change the combobox value while in this panel, the markers on the map are resized automatically according to the number of properties in the respective region. If there are no properties in a region for a selected price range, then no marker will be shown for that region.

If you hover the mouse over a neighborhood marker, the name of the neighborhood is displayed, <u>(windows doesn’t capture mouse pointer during screenshots so it is not therein the snapshot)</u>



From £: 25 To £: 1050

Price Range Selected: from £25 to £1050

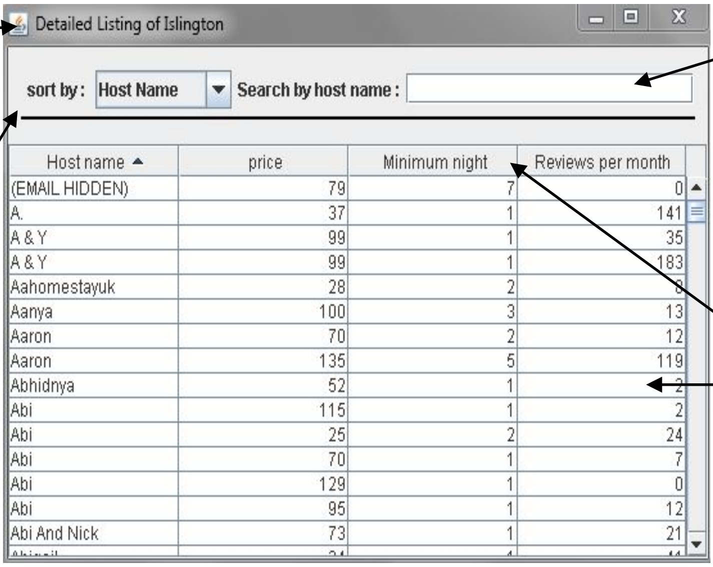
If invalid pair of data is selected in combo, then the user is taken back to the welcome screen. This is to ensure the map doesn’t display wrong/misleading data for an invalid input.

This pane has been made non-resizable because a user resizing this pane could mean some of the markers are no longer visible. If we made everything scale with resizing, that could cause some

The user can now navigate between the panels using the buttons, if there is no panel to navigate to, then the button is disabled

Data Listing window

Upon clicking a marker on the map screen, a new window is displayed with a list of all properties in the region, as shown below:



Detailed Listing of Islington

sort by: Host Name | Search by host name :

Host name ▲	price	Minimum night	Reviews per month
(EMAIL HIDDEN)	79	7	0
A.	37	1	141
A & Y	99	1	35
A & Y	99	1	183
Aahomestayuk	28	2	8
Aanya	100	3	13
Aaron	70	2	12
Aaron	135	5	119
Abhidnya	52	1	2
Abi	115	1	2
Abi	25	2	24
Abi	70	1	7
Abi	129	1	0
Abi	95	1	12
Abi And Nick	73	1	21

The region name of the marker clicked or the data being shown is shown in the title of the window. This pane is resizable

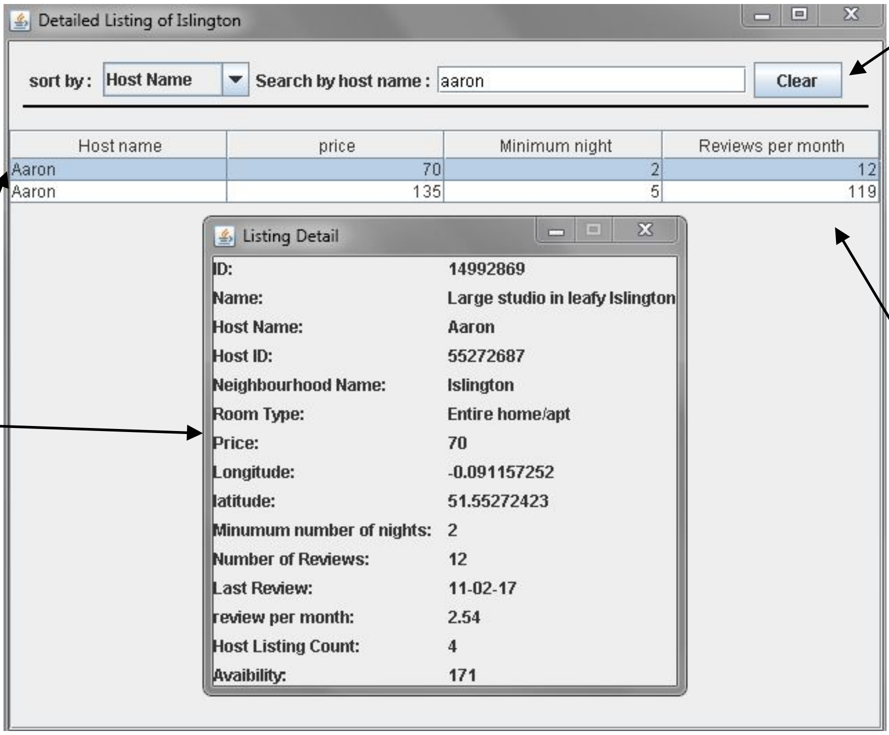
The user can sort the list by using this ComboBox, the user can sort by host name, price and number of reviews per month

The user can also search the list by inputting a Substring of any available hostname displayed in the list.

Clicking here will also sort it according to the column header

The user can click on any displayed rows in the list, and upon clicking the row, full details of the particular record in the row is displayed in a new window as illustrated below

Listing Details Window



Detailed Listing of Islington

sort by: Host Name | Search by host name : aaron | Clear

Host name	price	Minimum night	Reviews per month
Aaron	70	2	12
Aaron	135	5	119

Listing Detail

ID:	14992869
Name:	Large studio in leafy Islington
Host Name:	Aaron
Host ID:	55272687
Neighbourhood Name:	Islington
Room Type:	Entire home/apt
Price:	70
Longitude:	-0.091157252
latitude:	51.55272423
Minumum number of nights:	2
Number of Reviews:	12
Last Review:	11-02-17
review per month:	2.54
Host Listing Count:	4
Avaibility:	171

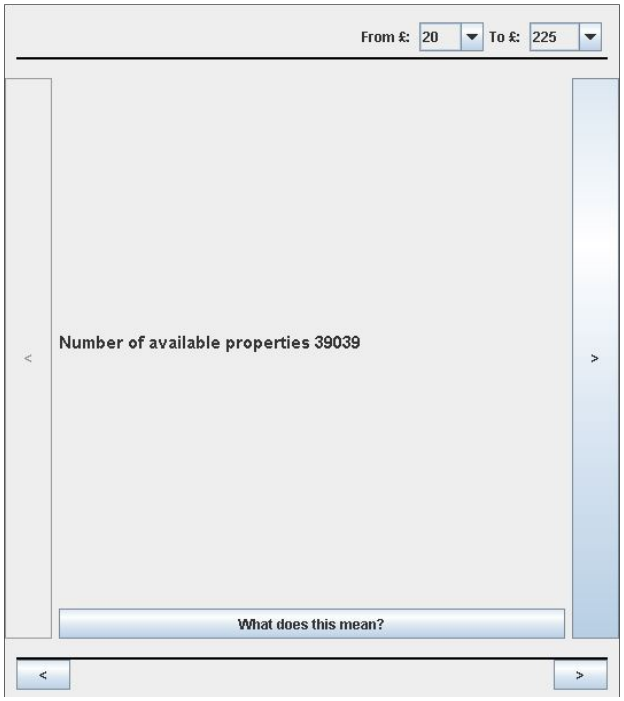
When you start searching for a record, the clear button is visible, so u can clear the search textbox. Upon clearing, the table is returned to its original stage

To illustrate, this particular record has been selected in the list, and upon clicking, a new window pops up, displaying full details of that particular record

Data listing filtered according to search text input in the textbox

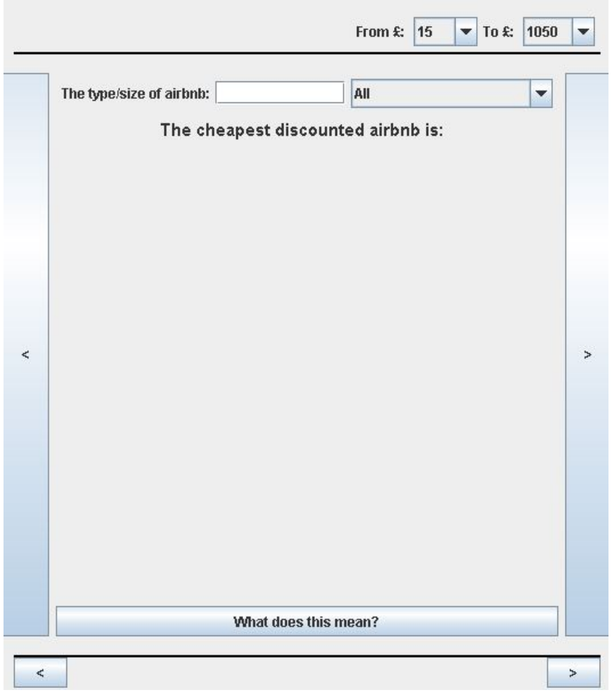
Statistics Panel

The statistics panel has distinguished navigation buttons on left and right for navigation between a total of 9 statistics, as shown in the snapshots below. If you have reached end of a statistic, then the navigation button for that end is greyed out, as shown in the picture on the left. A special statistics panel is designed to find cheapest discounted property, as shown in the image on the right. This panel requires some additional data for input, in order to provide a more specific statistic. **It is worth noting that all statistic panels are dynamically updated whenever the combo input is changed, in order to reflect statistics only for that particular range.** Therefore, if invalid combo input is selected, then the user is taken back for the instructions panel, to guide the user to select a valid combo pair. **Each panel has a button in the bottom, which upon clicking, displays what the statistic means.**



From £: 20 To £: 225

Number of available properties 39039



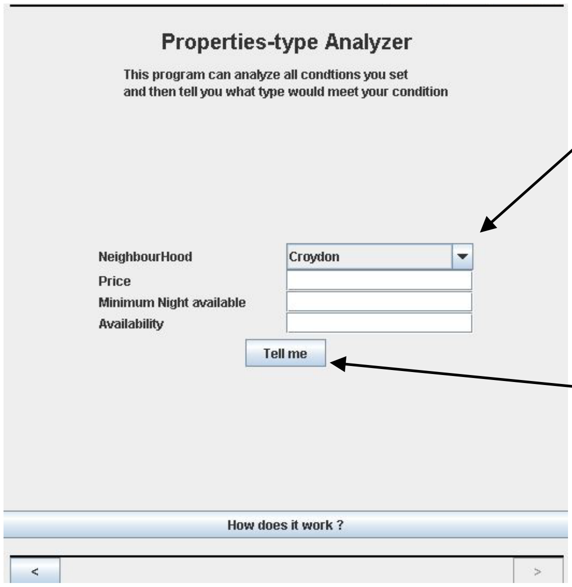
From £: 15 To £: 1050

The type/size of airbnb: All

The cheapest discounted airbnb is:

Final/Challenge Task Panel

In the initialization stage of this panel, the classifier gets trained from over 50000 records in the data file, using the "property type" column as target attribute. After training is complete, the algorithm can run completely independently of the original dataset making predictions at an accuracy of 90%.



Properties-type Analyzer

This program can analyze all condtions you set
and then tell you what type would meet your condition

NeighbourHood Croydon

Select neighborhood from dropdown, and the input the price, min, night and availability

After inputting data and clicking this button, the program will predict what type of property it might be based on your input, and it does so, by being completely independent from the data itself.

Extension task description:

For the challenge task, we used a machine learning and data mining API, to build a properties-type analyser. We have created a class known as the **classifier**, which concerns about setting up the functionality of the data mining API. Moreover, we created a custom exception class for its error handing. We extracted data needed to train the classifier from the csv file to an attribute relation format data file (.arff extension file named as “train_dataset.arff”). The classifier gets trained from over 50000 data instances in the file, using the rightmost column in the data file (the room type attribute) as target attribute. We used the J48 algorithm for training (more details below). **After training, it is ready to predict the property type based on given input of any range with a 90% accuracy, and it does so by being completely independent of the data. i.e. the prediction AI is not dependent on the data anymore. The implementation and justification for the algorithm type we used is given below:**

The properties-type analyzer or the classifier uses an implementation of C4.5 algorithm called J48 to build a decision tree. In the C4.5 algorithm, the attribute that come with the highest information gain for target attribute is selected as the (decision of) root node, then the root node is spilt into sub-nodes with branches representing different value of the attribute at root node (the value at each branches is a subset of the root node value) Then attributes for these sub-nodes can be computed by selecting the attribute that has the highest information gain for each subset of the root nodes. This process is repeated until 1) all attributes has appeared in a path. 2) the training example associated with this leaf node all have the same target attribute value[1]

Information gain can be computed from entropy. Entropy states how informative a random collection of examples is. Which can be expressed as following equation:

$$Entropy(S) = \sum_{x \in C} -p(x) log_2 p(x)$$

Where S is the dataset we are calculating entropy with.

C is the set of the target attribute (classes)

$p(x)$ is the proportion of the number of element in x to those of S.

Information gain is simply the expected reduction in entropy caused by partitioning the examples according to this attribute[2]. And It can be expressed as equation based on entropy:

$$Information\ Gain(S, \varphi) \equiv Entropy(S) - \sum_{x \in Value(\varphi)} \frac{|S_x|}{|S|} Entropy(S_x)$$

Where φ is the set of all value in attribute φ

and $S_x = \{a \in S \mid \varphi(a) = x\}$

So to summarize, this deep learning algorithm is no longer reliant on the data file. The contents in the .arff training data file were used to **train** the deep learning algorithm and increase its accuracy in making future predictions about property type. After training is complete, the algorithm can run **completely independently** of the original dataset.

[1] Mitchell, Tom M. (1997). *Machine Learning*, Mc-Graw Hill Education, p.60

[2] Mitchell, Tom M. (1997). *Machine Learning*, Mc-Graw Hill Education, p.57