Interactive Dashboard of Dublin Rental Properties and Predictive Model

Group Number - 27

Member details

Name	Email	Student ID
Aisling Keating	aisling.keating26@mail.dcu.ie	20213464
Anna Field	anna.field4@mail.dcu.ie	21269030
Colin Hehir	colin.hehir3@mail.dcu.ie	20213371
Ellen Jaye Woods	ellen.woods8@mail.dcu.ie	20213494
Niamh Eilis Feeney	niamh.feeney6@mail.dcu.ie	20213485
Thomas Feeney	thomas.feeney5@mail.dcu.ie	20213491

Section 1 - Dataset, Source Code for Git Repository and App

Dataset: https://www.kaggle.com/d17129765/predicting-dublin-rental-daftie

GitHub: colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)

Interactive Dashboard:
https://public.tableau.com/app/profile/aisling.keating/viz/CA675Group27DublinPropertyFinder/Model?publish=yes

Section 2 - Introduction and Motivation for the Application

Dublin Housing Crisis - Increasing Demand, Less Available Homes and Increasing Prices

The rise in property prices has been a prominent issue in Ireland over the past few years and with reports that this is to continue to rise it is clear that this is a very important issue to gain further insights into [1]. The Covid-19 pandemic has had an effect on both Dublin's rental and buying market, seeing prices surge and the number of houses available decreasing with reports showing that just 2,455 rental homes were available across the country on 1 August 2021, the lowest number since records began in 2006 [2] and with 50% less houses available to buy between 2020 and 2021 [3]. Reports on Daft.ie show that this unprecedented scarcity of rental homes caused the average national rents to climb by 5.6% [2] and house prices by 9% in one year [3]. The persistent imbalance between the housing supply and demand leads to affordability pressures for renters with house prices continuing to rise.

Reasons for Imbalance between Supply and Demand

There are many reasons behind this imbalance between supply and demand. Studies show that Ireland has had population growth and net migration even during the worst of the pandemic. Along with this there is strong evidence of suppressed household formation, therefore it is predicted that this high demand for housing is set to remain strong until well into the middle of the century [3]. There has also been some Pandemic-related disruption that has caused many building sites to close over the past year, this has put a hold on any apartments being built or renovated and more and more new builds release dates are being pushed back [3]. Another impact on increasing prices is that large institutional investors bulk purchase new builds to rent out, this is making it impossible for renters to become homeowners and is pushing more people into renting property and for longer, therefore having a follow on impact on the rental market [2].

Cuckoo Funds

These large institutional investors are known as ‘cuckoo funds’ or Private Rented Sector (PRS) funds and are backed by institutional investors like pension funds. These investments in Ireland started in 2013 and from then this has grown to €1.1 billion invested in almost 3,000 units last year [4]. A report from Savills states that between 2012 and 2018, block purchasers bought 9,291 units in Dublin City, accounting for 8.1% of all the residential properties that have been purchased [4]. This report from Savil noted that build-to-sell schemes were being entirely bought-out by these PRS investors during the construction phase due to ‘block sale’ of apartments removing risk and providing a quicker return on capital for the developers.

These PRS investors have been heavily criticised for causing Ireland’s housing crisis. They are said to be ‘gobbling up properties and squeezing first-time buyers out of the housing market’ by Fianna Fáil housing spokesperson Daragh O’Brien. This is where the term for

cuckoo funds has originated, comparing these funds to the birds that push the others' eggs out of their nests and move in [4].

The government has been heavily encouraged to urgently review the tax treatment of these funds as many are effectively operating tax-free while charging high levels of rent. These funds have been driving up rents for years as well as taking the properties out of the market, meaning first-time buyers are not getting their foot on the property ladder [4]. This has pushed more people into renting property as well as keeping these people renting for longer before they are able to buy, so is impacting on the country's rental market as well. This along with the tighter regulation of mortgages, the need for higher deposits and increasing prices of houses is keeping people renting [4].

New Legislation

With this Rental Market crisis, the Government has been prompted to bring in new legislation reviewing the tax treatment of these funds due to the fact many are effectively operating tax-free while charging high levels of rent [4]. In June 2021, the Government passed legislation aimed at disincentivizing these PRS funds from bulk purchasing new properties in Ireland [2]. This legislation includes planning and tax changes such that Stamp duty on the purchase of 10 or more houses was raised to 10 per cent, this was increased from 7.5% [5].

However there have been some controversial amendments to the legislation, passed June 2021, that allow these PRS funds to sidestep this new stamp duty rate of 10% when bulk purchasing more than ten homes if they plan to lease the properties back to the State for social housing [6].

Ireland's Targets to Satisfy Housing Needs

This has been a problem for many years, and has been worsened by the pandemic. Research into the housing area has predicted that there is a need of roughly 50,000 new homes annually, for every year between 2016 and 2051. This includes owner-occupied homes, private rental and social housing sectors housing [3]. To be on target for this, Ireland would have needed to have built 250,000 between 2016 and 2020, however, the country built fewer than 85,000 [3]. Between 2020 and 2023, as stated previously, the pandemic has also stunted the amount of housing that would have been needed to be built due to construction site closures and delays. This is putting further pressure on Ireland's already constrained housing supply, with only 60,000 homes expected to be built between 2020 and 2023 [2].

With these figures, it is evident that this is likely to remain an issue well into the future, hence we have decided to build an interactive tool that can show the different properties in Dublin by their respective attributes, such as price and description, through an interactive map and to build a model to predict price of properties price by selecting different variables. The aim of this is to help renters to find the properties that they can afford with their budget and to research the areas that these are likely found in.

Section 3 - Choice of the Technologies

As a result our proposed solution utilised Apache Pig for Data Cleaning and Processing: while our dataset was validated and verified through a completeness and accuracy check via Apache Hive. Our User Interface was then a Dashboard application produced via Tableau Public.

The reason we chose these technologies is as follows:

Google Cloud Platform (GCP)

GCP is a flexible, open, secure and cost effective way to utilize services such as DataProc as you can integrate your data lake meta store with open-source data clusters you build [7] [8] [9]. Therefore, we created a Hadoop Cluster on GCP to perform the subsequent tasks [15].

Apache Hadoop (MapReduce)

We chose to use Apache Hadoop as it is an open source software framework that is reliable, scalable and fault-tolerant. We used it to analyze and process the big data we had and to enable distributed computing of data in a cluster on Google DataProc [7]. The Apache Hadoop computer application is a concept that includes basic programming principles to enable the cloud computing of massive data volumes across multiple nodes. It's built to expand from a dedicated processor to thousands of devices, each with its own computing and storage capabilities [10]. Rather than relying on equipment to provide full functionality, the package is designed to identify and tackle issues at the application level, allowing a massively scalable service to be delivered on top of a network of machines that may all fail [16].

Apache Pig

Pig was utilized to perform ETL and data processing on the data collected. Its ability to perform this made it a vital tool for this project [7]. It is a data analysis infrastructure that comprises a high-level framework for writing data analysis algorithms and architecture for

assessing these programs [11]. Pig programs are notable for their design, which allows for significant parallelization and, as a result, the handling of very big data [17].

Apache Hive

We used Hive to get specific answers from the respective dataset as it supports SQL-style queries and works off a structured schema built around tables, rows, columns and queries [7] [10]. Using HiveQL, the data storage technology makes it easier to view, write, and manage big datasets stored in cloud databases. Data that has previously been stored can be therefore transferred with structure [7] [12] [13] [18].

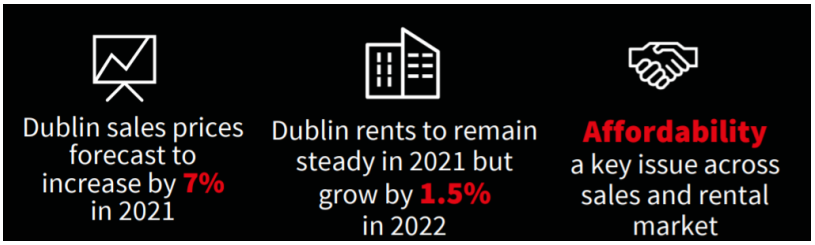
Tableau

Using the data visualisation, business intelligence and analytics software capabilities of Tableau, we built an interactive map as well as a dashboard to show Key Performance Indicators at a glance. The Tableau software, which is the market-leading solution for advanced business analysis, is known for rapidly and effortlessly turning any type of information from practically any source into actionable information. Drag and drop is all there is to it. In addition, the sector engagement resources, training, and global data network provide unmatched support to clients' analytics investments. In addition, as part of their mission to help people see and interpret data, they go above and beyond with their technology to assure client satisfaction by assisting employees in developing a performance - based culture. Tableau Public was used to build our User Interface as it is a free software that allows anyone to access a database or file and generate web-based interactive data representations [19].

Section 4 - Related Work

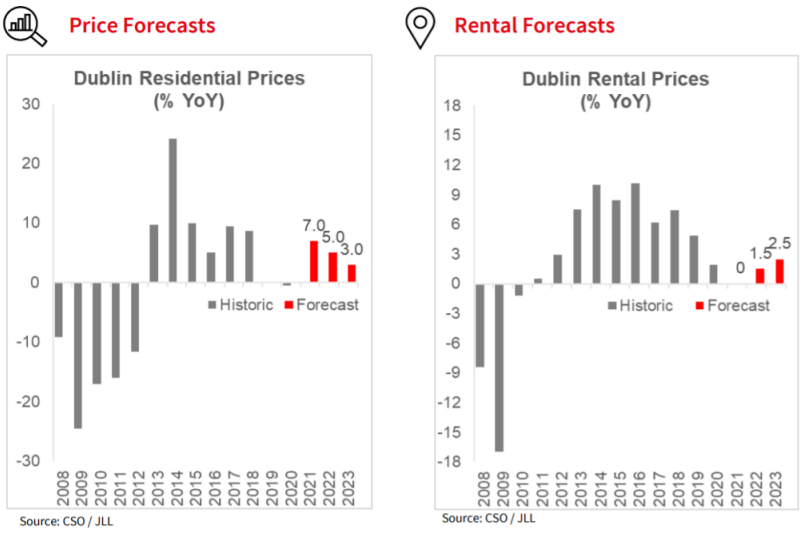
JLL - Bi-Annual Dublin Residential Forecasts Report for H2 2021

JLL, a real-estate advisors and professionals firm based in Ireland, published some interesting trends and insights relating to the Dublin residential property market.



Their residential forecasting model uses 15 economic and real estate market parameters to provide a three-year projection for rentals and rates. The complicated model additionally takes into account industry knowledge and opinion in order to determine and evaluate the results [20].

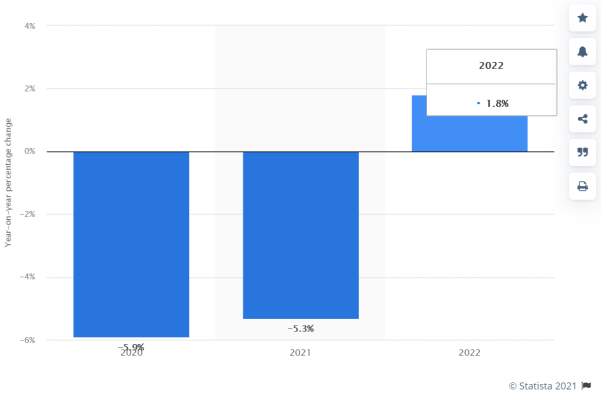
Real estate prices in Dublin are expected to rise by 7% in 2021, 5% in 2022, and 3% in 2023, according to projections. Changes to rent price of 0% in 2021, 1.5 percent in 2022, and 2.5 percent in 2023 are also expected [20].



Statistica - Forecast of the Percentage Change on the Previous Year of Residential Property Prices in Ireland from 2020 to 2022

Statistica provides reliable market and business data as well as an advanced analytics software package, while also being considered a market leader in this space. The research shows house prices in the Republic of Ireland are expected to fall after COVID-19, before increasing once more in 2022, according to projections. According to Statistica, this view is

more encouraging than it was previously in 2020, when housing market declines of up to 12% were projected [21].



Section 5 - Description of the Dataset

Source of the Data

After careful consideration of multiple potential datasets, we chose a dataset from Kaggle of the rent prices in Dublin taken from Daft.ie in September 2020, as after some exploratory analysis it was found that it contained the required information to fulfil the scope of this assignment.

We considered this as a ‘big data’ dataset as the source it was scraped from, Daft.ie, has the following characteristics of big data:

Volume	1000s of houses/apartments/properties advertised for sale or rent.
Velocity	100s of advertised properties added daily.
Variety	Data coming from different sources (landlords) and types (text, pictures etc.)
Veracity	All users adding data must be verified to be a real person before placing an ad.

Process of Extraction/Collection

The dataset was scraped by GuanLong Lyu [22] in September 2020 from Daft.ie, Ireland's No.1 Property website and app, which he provided for public use on Kaggle.com. Due to this information being taken manually from a website, it was presumed the owner considered the site’s terms and conditions and spaced out web requests. While the individual may have reviewed the collected data for cleaning, we decided to carry out further cleansing procedures detailed in the next section. After saving down the dataset as a .csv file, we noted its following characteristics:

Metadata of Sourced Dataset

price	address	bathroom	bedroom	furnish	description	property type	ID	longitude	latitude
â,¬8,400 P	Sorrento R	5	5	Furnished		House	21612293	-6.096875	53.27431
â,¬15,000	Ailesbury R	6	6	Furnished		House	22045922	-6.217086	53.32006
â,¬15,000	5 Elgin Roa	5	5	Unfurnish		House	22048233	-6.236891	53.33003
â,¬10,000	Elgin Road,	5	4	Unfurnish		House	22043562	-6.233272	53.32961
â,¬9,400 P	3 Tempe Te	5	3	Unfurnish		House	22059358	-6.100353	53.27699
â,¬8,950 P	Tivoli Terra	6	4	Furnished		House	21920497	-6.139981	53.28957
â,¬8,500 P	Alexander	4	4	Furnished		Apartment	21916321	-6.246952	53.34127
â,¬8,500 P	South Circu	9	5	Furnished		House	22057032	-6.286758	53.3324
â,¬8,000 P	Malakoff V	5	4	Furnished		House	22043386	-6.267877	53.32026
â,¬8,000 P	Belmont Av	4	4	Unfurnish		House	22004847	-6.242893	53.32003
â,¬1,800 P	22 Mountjo	7	4	Furnished		House	22016581	-6.259054	53.35667

Size	2,678KB (as a .csv file)	**Instances**	2718

Attribute									
Price	Address	Bathroom	Bedroom	Furnish	Description	Property Type	ID	Latitude	Longitude
Data Type									
Ratio	Nominal	Ratio	Ratio	Nominal	Nominal	Nominal	Ratio	Interval	Interval

Section 6 - Data Preprocessing – Preparation and Cleaning

Environment Variables Set Up in GCP Hadoop Cluster

Firstly a Google Cloud Platform account was created where a cluster for this assignment was set up via the Dataproc service. After accessing the Command Line Interface, the following code was run to check our username.

```
$ whoami
```

Next we set up the Hadoop Distributed File System in the home directory for our current user.

```
$ hadoop fs -mkdir -p /user/colin_hehir3
```

To not have to set up the environment variables for every time we opened a new SSH terminal, we set up JAVA_HOME, PATH, and HADOOP_CLASSPATH with the following after ensuring we had the required java version.

```
$ java -version

$ export PATH=${JAVA_HOME}/bin:${PATH}
```


```
$ export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

In order to see that there was no error and to make sure the cluster was successfully set-up, the following command was run to show the list in the HDFS home directory.

```
$ hadoop fs -ls /
```

Lastly to ensure that the environment variables are set, the following command was run.

```
$ env
```

Load Data from Files into HDFS

To load our selected dataset into HDFS, we firstly created a directory within the environment.

```
$ hadoop fs -ls /

$ hadoop fs -mkdir /Group_27_Data_Acquisition
```

After, we uploaded the respective file through selecting ‘Settings>Upload file’. We then used the following command to move the file from the home directory to one created in HDFS, before listing out the files to ensure the transfer was successful.

```
$ hadoop fs -put /home/colin_hehir3/Daft_Dublin_Rent_Sept_2020_Dataset.csv
/Group_27_Data_Acquisition

$ hadoop fs -ls /Group_27_Data_Acquisition
```

Pig ETL - Load Dataset and Column Filter

Pig was initialised and the .csv raw dataset was loaded using the org.apache.pig.Piggybank. storage.CSVExcelStorage function, while taking into careful consideration of the type of delimiter used, data type selected and amount of fields we had. A sense-check was then completed by creating a mini dataset of five instances and using the dump command to visually see the data efficiently.

```
$ pig

grunt> Daft_Dublin_Rent_Sept_2020 = LOAD
'/Group_27_Data_Acquisition/Daft_Dublin_Rent_Sept_2020_Dataset.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','YES_MULTILINE','NOCHA
NGE','SKIP_INPUT_HEADER') AS(Price:chararray, Address:chararray,
Bedroom:int, Bathroom:int, Furnish_Unfurnished:chararray,
Description:chararray, Property_Type:chararray, ID:int,
Longitude:chararray, Latitude:chararray);
```


```
grunt> TestData_Daft_Dublin_Rent_Sept_2020 = LIMIT
Daft_Dublin_Rent_Sept_2020 5;

grunt> DUMP TestData_Daft_Dublin_Rent_Sept_2020
```

Due to some of the fields being identified as redundant and not needed for the rest of the project, we applied a sort of column filter using the ForEach and Generate commands to select the ones required. Again, a sense check was completed to ensure this had been conducted correctly.

```
grunt> Daft_Dublin_Rent_Sept_2020_Column_Filter = FOREACH
Daft_Dublin_Rent_Sept_2020 GENERATE
Price,Address,Bedroom,Bathroom,Furnish_Unfurnished,Property_Type,Longitude,
Latitude;

grunt> TestData_Daft_Dublin_Rent_Sept_2020_Column_Filter = LIMIT
Daft_Dublin_Rent_Sept_2020_Column_Filter 5;

grunt> DUMP TestData_Daft_Dublin_Rent_Sept_2020_Column_Filter
```

Pig ETL - Data Cleaning - Removing Data Artefacts and Replacing Null Values

The initial dataset presented many challenges, one being that there were many data artefacts and null values that had to be removed. This was likely due to it being scraped from a website where this result is very common in the process.

Due to the functionality in Daft.ie, it is not required to input if a property is furnished or not. This resulted in many properties being marked as ‘Furnished or unfurnished’ as well as being left blank altogether.

```
grunt>
Daft_Dublin_Rent_Sept_2020_Furnished_Unfurnished_Populate_Unspecified_Value
s = FOREACH Daft_Dublin_Rent_Sept_2020_Column_Filter GENERATE
Price,Address,Bedroom,Bathroom,(Furnish_Unfurnished == 'Furnished or
unfurnished' ? 'Unspecified' : Furnish_Unfurnished) AS
Furnish_or_Unfurnished,Property_Type,Longitude,Latitude;

grunt>
Daft_Dublin_Rent_Sept_2020_Furnished_Unfurnished_Populate_Null_Values =
FOREACH
Daft_Dublin_Rent_Sept_2020_Furnished_Unfurnished_Populate_Unspecified_Value
s GENERATE Price,Address,Bedroom,Bathroom,(Furnish_or_Unfurnished == '' ?
'Unspecified' : Furnish_or_Unfurnished) AS
Furnish_or_Unfurnished,Property_Type,Longitude,Latitude;
```

We identified that the Daft.ie properties listing may include multiple adverts for the same property (e.g. reuploaded every week or so). We therefore decided to remove any duplicate properties by their address.

```
grunt> Daft_Dublin_Rent_Sept_2020_Grouped = GROUP
Daft_Dublin_Rent_Sept_2020_Furnished_Unfurnished_Populate_Null_Values BY
Address;

grunt> Daft_Dublin_Rent_Sept_2020_Remove_Duplicates = FOREACH
Daft_Dublin_Rent_Sept_2020_Grouped {result = TOP(1, 0, $1);GENERATE
FLATTEN(result);}
```

There were a few properties who did not have any bathrooms or bedrooms. For our project and eventual model, we decided to only consider properties who had these as a prerequisite. Therefore we removed any line items with zero bathrooms or bedrooms.

```
grunt> Daft_Dublin_Rent_Sept_2020_Remove_No_Bedrooms = FILTER
Daft_Dublin_Rent_Sept_2020_Remove_Duplicates BY Bedroom !=0;

grunt> Daft_Dublin_Rent_Sept_2020_Remove_No_Bathrooms = FILTER
Daft_Dublin_Rent_Sept_2020_Remove_No_Bedrooms BY Bathroom !=0;
```

Similarly after exploring the data it was identified that the Longitude and Latitude fields had unwanted symbols such as commas and quotation marks as well as other random data artefacts. The properties that contained these were also removed, on the presumption that it did not increase bias and help to prevent skewing.

```
grunt> Daft_Dublin_Rent_Sept_2020_Remove_Longitude_Data_Artefacts = FILTER
Daft_Dublin_Rent_Sept_2020_Remove_No_Bathrooms BY NOT
(ENDSWITH(Longitude,',') OR ENDSWITH(Longitude,'"') OR
ENDSWITH(Longitude,'sel') OR ENDSWITH(Longitude,'s'));

grunt> Daft_Dublin_Rent_Sept_2020_Remove_Latitude_Data_Artefacts = FILTER
Daft_Dublin_Rent_Sept_2020_Remove_Longitude_Data_Artefacts BY NOT
(ENDSWITH(Latitude,',') OR ENDSWITH(Latitude,'"') OR
ENDSWITH(Latitude,'sel') OR ENDSWITH(Latitude,'s'));
```

Our last data cleaning protocol was to remove the Euro sign from the price column as well as the thousands separator comma. This was to aid us to conduct calculations and create new columns by having just a number with only digits.

```
grunt> Daft_Dublin_Rent_Sept_2020_Remove_Euro_Comma_Characters = FOREACH
Daft_Dublin_Rent_Sept_2020_Remove_Latitude_Data_Artefacts GENERATE
REPLACE(REPLACE(Price,'(€)',''),'(,)','') AS
Price,Address,Bedroom,Bathroom,Furnish_or_Unfurnished,Property_Type,Longitu
de,Latitude;
```

Pig ETL - Data Processing - Splitting Columns and Attributes

To provoke more analysis and modelling down the line, we explored the possibility of creating new attributes or fields in our dataset. In order to do this we looked at the Price

column where properties had a price of either per week or per month. We therefore created a new column where the data only included the value from the price column.

```
grunt> Daft_Dublin_Rent_Sept_2020_Split_Price_Column = FOREACH
Daft_Dublin_Rent_Sept_2020_Remove_Euro_Comma_Characters GENERATE
REPLACE(REPLACE(Price,' Per week',''),' Per month','') As
Value,Price,Address,Bedroom,Bathroom,Furnish_or_Unfurnished,Property_Type,L
ongitude,Latitude;

grunt> Daft_Dublin_Rent_Sept_2020_Split_Week_Or_Month_Column = FOREACH
Daft_Dublin_Rent_Sept_2020_Split_Price_Column GENERATE
Value,REPLACE(REPLACE(Price,'.*Per week','Per week'),'.*Per month','Per
month') As
Week_Month,Price,Address,Bedroom,Bathroom,Furnish_or_Unfurnished,Property_T
ype,Longitude,Latitude;
```

Next we looked at the Address column. Since all addresses were unique, we couldn't segment our properties into different areas. However, using the last section of the address column, we created a new area attribute which assigned a property into one of seven distinct areas around Dublin.

```
grunt> Daft_Dublin_Rent_Sept_2020_Area_Column = FOREACH
Daft_Dublin_Rent_Sept_2020_Split_Week_Or_Month_Column GENERATE
Value,Week_Month,Price,REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPL
ACE(Address,'.*South Co. Dublin','South Co. Dublin'),'.*South Dublin
City','South Dublin City'),'.*Dublin City Centre','Dublin City
Centre'),'.*North Co. Dublin','North Co. Dublin'),'.*North Dublin
City','North Dublin City'),'.*West Co. Dublin','West Co. Dublin'),'.*Co.
Dublin','Co. Dublin') As
Area,Address,Bedroom,Bathroom,Furnish_or_Unfurnished,Property_Type,Longitud
e,Latitude;
```

This was then stored into our Data Cleaning folder using the org.apache.pig.Piggybank.storage.CSVExcelStorage function.

```
grunt> STORE Daft_Dublin_Rent_Sept_2020_Area_Column INTO '/DataCleaning'
USING org.apache.pig.piggybank.storage.CSVExcelStorage();
```

This was then saved in the file path '/DataCleaning/part-m-00000'.

Pig ETL - Data Creation - Calculations and New Attributes

Our saved dataset was reloaded into Pig, but instead using different data types such as int, double and bigdecimal in order to carry out calculations and create new fields.

```
grunt> Daft_Dublin_Rent_Sept_2020_Cleaned = LOAD
'/DataCleaning/part-m-00000' USING
org.apache.pig.piggybank.storage.CSVExcelStorage()
AS(Value:int,Week_Month:chararray,Price:chararray,Area:chararray,Address:ch
ararray,Bedroom:double,Bathroom:double,Furnish_or_Unfurnished:chararray,Pro
perty_Type:chararray,Longitude:bigdecimal,Latitude:bigdecimal);
```


The value field was updated where if the value was initially per week, it was multiplied by four in order to have the equivalent of per month and therefore making the whole value common and comparable with each instance in it.

```
grunt> Daft_Dublin_Rent_Sept_2020_Price_Per_Month_Column = FOREACH
Daft_Dublin_Rent_Sept_2020_Cleaned GENERATE (Week_Month == 'Per week' ?
Value*4:Value) AS
Price_Per_Month,Value,Week_Month,Price,Area,Address,Bedroom,Bathroom,Furnis
h_or_Unfurnished,Property_Type,Longitude,Latitude;
```

Next, a new field was created which is essentially the price per month per bed. We felt this was a better attribute to look at and compare to different areas around Dublin as a one bedroom house would always be inferior in price compared to a four bedroom property for example.

```
grunt> Daft_Dublin_Rent_Sept_2020_Price_Per_Month_Per_Bed_Column = FOREACH
Daft_Dublin_Rent_Sept_2020_Price_Per_Month_Column GENERATE
Price_Per_Month,(Price_Per_Month/Bedroom) AS
Price_Per_Month_Per_Bed,Address,Area,Bedroom,Bathroom,Furnish_or_Unfurnishe
d,Property_Type,Longitude,Latitude;
```

From our experience, having a house where there was a low amount of bathrooms shared would always be in demand and desirable to rent. We therefore created a new column called the bathroom to bedroom ratio where it essentially gave the number of how many bathrooms there would be to one person. This would be beneficial to include in our interactive dashboard in order for a user to select a threshold in how low a ratio a property has that he or she would be willing to rent in.

```
grunt> Daft_Dublin_Rent_Sept_2020_Bathroom_To_Bedroom_Ratio_Column =
FOREACH Daft_Dublin_Rent_Sept_2020_Price_Per_Month_Per_Bed_Column GENERATE
Price_Per_Month,Price_Per_Month_Per_Bed,(Bathroom/Bedroom) As
Bathroom_To_Bedroom_Ratio,Address,Area,Bedroom,Bathroom,Furnish_or_Unfurnis
hed,Property_Type,Longitude,Latitude;
```

Our last command in this section was to rename this dataset to a final version before storing it and begin to work on our model and interactive dashboard.

```
grunt> Daft_Dublin_Rent_Sept_2020_Final = FOREACH
Daft_Dublin_Rent_Sept_2020_Bathroom_To_Bedroom_Ratio_Column GENERATE
Price_Per_Month,Price_Per_Month_Per_Bed,Bathroom_To_Bedroom_Ratio,Address,A
rea,Bedroom,Bathroom,Furnish_or_Unfurnished,Property_Type,Longitude,Latitud
e;
```

Pig ETL - Store New Dataset

After this ETL process, the dataset was stored as a .csv file where the HDFS directory of the file was listed to be sure if it was saved in the correct location.

```
grunt> STORE Daft_Dublin_Rent_Sept_2020_Final INTO
'/Daft_Dublin_Rent_Sept_2020_Final_Dataset' USING
org.apache.pig.piggybank.storage.CSVExcelStorage('|','NO_MULTILINE','NOCHAN
GE','SKIP_OUTPUT_HEADER');

$ hadoop fs -ls /Daft_Dublin_Rent_Sept_2020_Final_Dataset
```

This was then saved in the file path '/Daft_Dublin_Rent_Sept_2020_Final_Dataset/ part-m-00000'.

Section 7 - Data Processing – Validation

Hive - Table Creation and Data Load

In order to validate our cleaned and processed dataset from Apache Pig, we used Apache Hive in order to complete verification checks to ensure completeness and accuracy with zero errors.

Hive was initialised and a table was created to include fields of decimal (with two decimal places) and string respectively.

```
$ hive

hive> CREATE TABLE DUBLIN_RENTAL_PROPERTIES (Price_Per_Month
decimal(38,2),Price_Per_Month_Per_Bed
decimal(38,2),Bathroom_To_Bedroom_Ratio decimal(38,2),Address string,Area
string,Bedroom int,Bathroom int,Furnish_or_Unfurnished string,Property_Type
string,Longitude decimal(38,6),Latitude decimal(38,6)) ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|';
```

The cleansed data was then loaded into the newly created Dublin Rental Properties table.

```
hive> LOAD DATA INPATH
'/Daft_Dublin_Rent_Sept_2020_Final_Dataset/part-m-00000' INTO TABLE
DUBLIN_RENTAL_PROPERTIES;
```

Hive Queries - Completeness and Accuracy Verification and Data Validation

In order for us to identify fields and attributes the header function was set while we also gained valuable information through the Describe command. This gave us metadata about our loaded table such as data types, list of columns and its respective location.

```
hive> SET hive.cli.print.header=true;

hive> DESCRIBE DEFAULT.DUBLIN_RENTAL_PROPERTIES;
```

The Count command verified that our table had 1763 line items, or instances.

```
hive> SELECT COUNT(*) FROM DUBLIN_RENTAL_PROPERTIES;
```

Using Order By and Limit, we queried the top 10 and bottom 10 properties by their price per month, which was in line with our manually cleaned dataset.

```
hive> SELECT * FROM DUBLIN_RENTAL_PROPERTIES ORDER BY Price_Per_Month DESC
LIMIT 10;

hive> SELECT * FROM DUBLIN_RENTAL_PROPERTIES ORDER BY Price_Per_Month ASC
LIMIT 10;
```

To verify there were no data artefacts included in some columns, we used the Distinct command while also using Count to see how many instance types there were for Area, Property Type and if it was Furnished or Unfurnished. There were seven (South Co. Dublin, South Dublin City, Dublin City Centre, North Co. Dublin, North Dublin City, West Co. Dublin and Co. Dublin) distinct areas, four (apartment, house, flat and studio) property type and three (furnished, unfurnished and unspecified) specifications for the Furnished or Unfurnished column.

```
hive> SELECT COUNT (DISTINCT Area) FROM DUBLIN_RENTAL_PROPERTIES;

hive> SELECT COUNT (DISTINCT Property_Type) FROM DUBLIN_RENTAL_PROPERTIES;

hive> SELECT COUNT (DISTINCT Furnish_or_Unfurnished) FROM
DUBLIN_RENTAL_PROPERTIES;
```

The SUM was looked at for the price per month attribute, where it gave the value of €4,045,508 while the average price per month for a property according to this dataset was €2,294.67.

```
hive> SELECT SUM(Price_Per_Month) FROM DUBLIN_RENTAL_PROPERTIES;

hive> SELECT AVG(Price_Per_Month) FROM DUBLIN_RENTAL_PROPERTIES;
```

Similarly, looking at the price per month per bed attribute, the total value accumulated to €2,022,117.38 and the average was €1,146.98.

```
hive> SELECT SUM(Price_Per_Month_Per_Bed) FROM DUBLIN_RENTAL_PROPERTIES;

hive> SELECT AVG(Price_Per_Month_Per_Bed) FROM DUBLIN_RENTAL_PROPERTIES;
```

Lastly, we took an average of how many bathrooms there was to every bedroom which was 0.76907, or 77%.

```
hive> SELECT AVG(Bathroom_To_Bedroom_Ratio) FROM DUBLIN_RENTAL_PROPERTIES;
```


Metadata of Final Cleaned Dataset

Price Per Month	Price Per Month - Per Bed	Bathroom to Bedroom Ratio	Address	Area	Bedroom	Bathroom	Furnish/Unfurnished	Property Type	Longitude	Latitude
33600.00	6720.00	1	Sorrento	South	5	5	Furnished	House	-6.096875	53.274311
15000.00	2500.00	1	Ailesbury	South	6	6	Furnished	House	-6.217086	53.320058
15000.00	3000.00	1	5 Elgin R	South	5	5	Unfurnished	House	-6.236891	53.330032
10000.00	2000.00	0.8	Elgin Roa	South	5	4	Unfurnished	House	-6.233272	53.329613
9400.00	1880.00	0.6	3 Tempe	South	5	3	Unfurnished	House	-6.100353	53.276994
8950.00	1491.67	0.67	Tivoli Ter	South	6	4	Furnished	House	-6.139981	53.289568
8500.00	2125.00	1	Alexande	Dubli	4	4	Furnished	Apartment	-6.246952	53.341271
8500.00	944.44	0.56	South Cir	South	9	5	Unspecified	House	-6.286758	53.332395
8000.00	1600.00	0.8	Malakoff	South	5	4	Furnished	House	-6.267877	53.320257
8000.00	2000.00	1	Belmont	South	4	4	Unfurnished	House	-6.242893	53.320027
7200.00	1028.57	0.57	22 Moun	Dubli	7	4	Furnished	House	-6.259054	53.356674
7750.00	1550.00	1	Westmin	South	5	5	Unfurnished	House	-6.173972	53.269267
7500.00	1500.00	0.6	Currabin	South	5	3	Furnished	House	-6.113289	53.258829
7500.00	2500.00	1	Apartme	South	3	3	Furnished	Apartment	-6.093661	53.275072

Size	147KB (as a .xlsx file)	**Instances**	1763

Attribute										
Price Per Month	Price Per Month - Per Bed	Bathroom to Bedroom Ratio	Address	Area	Bedroom	Bathroom	Furnish/ Unfurnished	Property Type	Longitude	Latitude
Data Type										
Ratio	Ratio	Ratio	Nominal	Nominal	Ratio	Ratio	Nominal	Nominal	Interval	Interval

Data Storage and Data Source Creation

Using the Google Cloud Platform, we created a bucket where our cleaned and processed dataset from Apache Hive was created as a .csv file.

```
gsutil du -s gs://ca675_dublin_rental/

gsutil cp Daft_Dublin_Rent_Sept_2020_Final_Dataset.csv
gs://ca675_dublin_rental/
```

This therefore enabled us to access the dataset via Google BigQuery. We then downloaded the file from our GCP Storage Bucket to our local machine and uploaded it to our shared project folder on Google Drive for ease of access with our data visualisation tool. Since our data was static and would not change in the intervening time of the project, we felt having a connection with Google Drive was more efficient than having a live connection with Google BigQuery.

Section 8 - Data Model

A simple data model was created to predict the price of rent using the dataset. The code is below. The model takes a file as input that contains the amount of bedrooms, bathrooms and the location of the property. The output is the predicted price per month.
The idea behind the model is that the similarity and proximity of two properties are the best indicators that the cost of rent will be similar. This comes from the concept of minimising the euclidean distance between two points.

```
dataset = LOAD 'Daft_Dublin_Rent_Sept_2020_Final.csv' USING
org.apache.pig.piggybank.storage.CSVExcelStorage(',','YES_MULTILINE','NOCHA
NGE','SKIP_INPUT_HEADER') AS(Price:bigdecimal, PPR:chararray, BtB:int,
furn:int, Location:chararray, Bedroom:int, Bathroom:int, ID:int,
Type:chararray, Longitude:bigdecimal, Latitude:bigdecimal);

/*normalise and transform the dataset to include relevant datafields*/
dataset_n = FOREACH dataset GENERATE $0, $1, $5, $6, $9*-1, $10;

/*load the description of the property*/
property = LOAD 'Property.csv' using PigStorage(',') AS (beds:int,
baths:int, long:bigdecimal, lat:bigdecimal);

/*filter dataset to include properties with same amount of bathrooms and
bedrooms*/
prop_bed = FILTER dataset_n BY Bedroom == property.beds;
prop_bed_bath = FILTER prop_bed BY Bathroom == property.baths;

/*create values for a maximum distance from the location of the property
and filter our properties not within this distance*/
prop_n = FOREACH property GENERATE $0, $1, $2*-1, $3;
prop_min = FOREACH prop_n GENERATE $0, $1, $2-0.01, $3-0.01;
prop_max = FOREACH prop_n GENERATE $0, $1, $2+0.01, $3+0.01;
prop_f1 = FILTER prop_bed_bath BY $4 > prop_min.$2;
prop_f2 = FILTER prop_f1 BY $5 > prop_min.$3;
prop_f3 = FILTER prop_f2 BY $4 < prop_max.$2;
prop_f4 = FILTER prop_f3 BY $5 < prop_max.$3;

/*find the average price per month of the remaining properties and print
this as the predicted price per month*/
find_price = GROUP prop_f4 BY $2;
price_1 = FOREACH find_price GENERATE group, AVG(prop_f4.$0);
price = FOREACH price_1 GENERATE $1;
DUMP price;
```


- The dataset is first normalised, Dublin's longitude is negative for ease of calculation later this is made positive.
- The property data is then loaded into pig.
- The dataset is filtered so that it now only contains properties that are like the input property. In this case filtering so that only properties with the same number of bedrooms and bathrooms remain.
- The input property data is also normalised as above.
- Next the dataset is filtered to contain only properties that are near the input property. This is done by creating latitude and longitude limits around the input property and filtering out properties outside these limits.
- The average price per month of these properties is then calculated and is the output.

In order to validate the model, known properties were input and the predicted rent was compared to the actual rent. The results were good and rents were predicted quite closely to actual values.

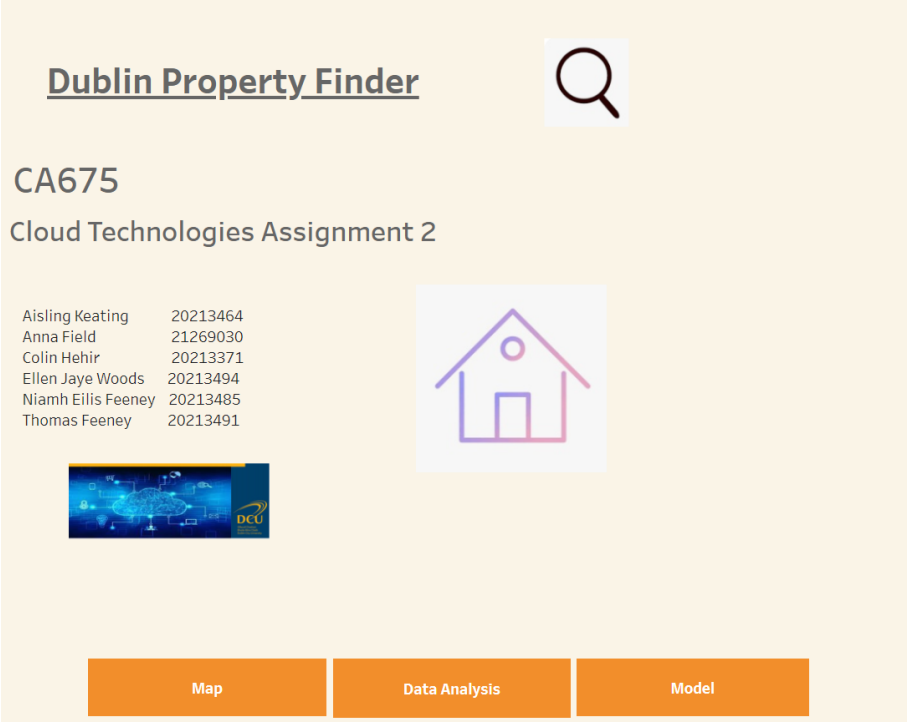
Description	Actual Rent	Predicted Rent	Difference
2 bed, 1 bath, Clyde Road	€2,100.00	€2,135.75	€35.75
2 bed, 1 bath, Appian Way	€2,000.00	€2,078.28	€78.28

To illustrate the model the rent for a property with 2 bedrooms and 1 or 2 bathrooms was found in various locations around Dublin. See the below section to view the output of this.

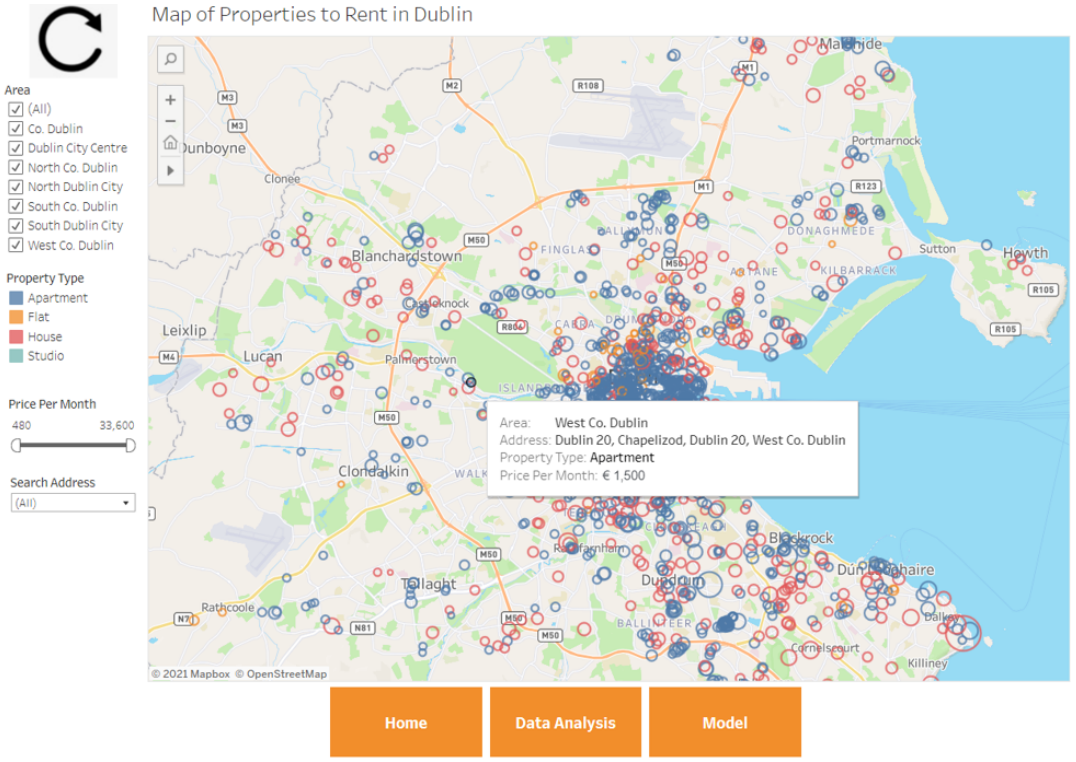
Section 9 - Development of the Application Platform

The application platform consists of four dashboard pages in Tableau so a user can navigate around to browse and get a prediction of rental properties in Dublin. It consists of the Homepage where the user can navigate to any of the other three pages. The Interactive Map page where the user can filter by area, price, property type and address to view available properties to rent around Dublin. You can also drill down by area and view more information of a property when the mark is selected. The price and type of the property can be easily distinguished by the size and colour of the marks on the map.There is also a Refresh button so the user can easily reset all the filters selected when browsing. The Data Analysis page is a summarized view of some KPIs which can help the user make an informed decision. The user can also drill down into the filters on this page. The Model page displays the predicted prices of 2 bed, 2 bath properties in Dublin by area. You can filter by area, price and also by viewing the different marks on the map. The user can easily view the more expensive properties as they are larger in size and coloured darker. This workbook is published online so any user can access and browse it.

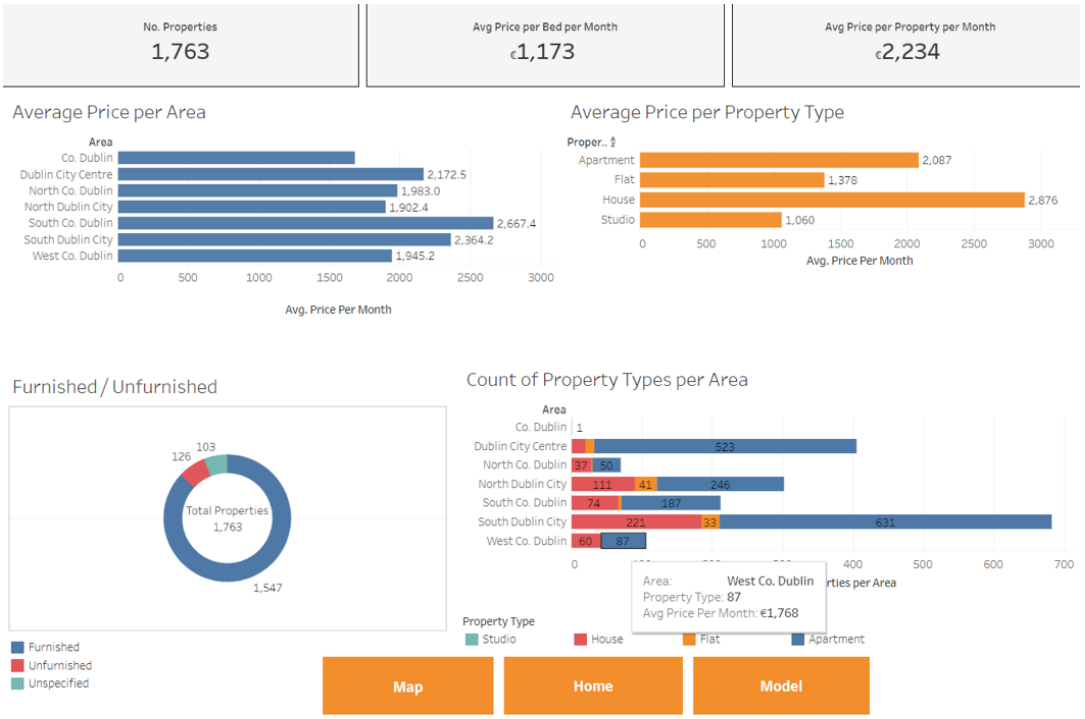
Homepage of User Interface/Interactive Dashboard



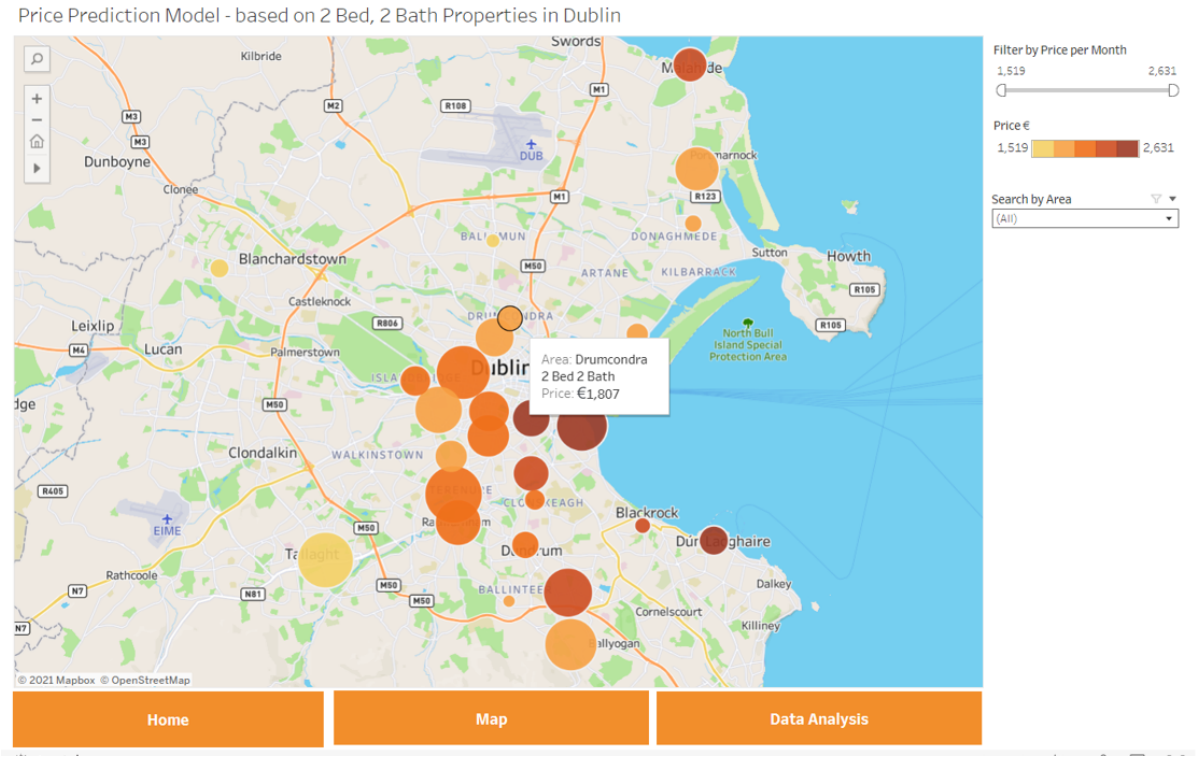
Interactive Map



Data Analysis KPI Dashboard



Interactive Model - Selecting based on Area and Price



Section 10 - Challenges and Lessons Learned

Completing this assignment provided a steep learning curve where the team gained valuable experience and skills in relation to distributed cloud computing and UX design as well as refining soft skills that will become invaluable at the workplace, or indeed in research in the future. However, it also presented many challenges that the group had to overcome. Since all the group are completing the Masters in Computing part-time, we had to be very economical and diligent with our schedule, where time management was a key factor in getting this project completed and up to standard in the specified time frame. Due to people being busy with work and life commitments, it was difficult to keep track of individual tasks being conducted. As a result, we were dynamic and flexible with our internal deadlines for each task, where everyone completed their work in their own time, while also maintaining a high standard of output and deliverables. We conducted catch-ups on a weekly basis and also created a group chat via a mobile application for ease of communication to keep the team updated. On reflection, it would have been more beneficial to increase the frequency of these meetings to two or three times a week in order to keep the project moving at a relatively fast pace.

Technically, the team had no previous experience with Apache Hadoop, Pig or Hive as well as Google Cloud Platform. While it was a challenge to get up to speed with the basics of these technologies, through teamwork, coaching and perseverance we were able to produce the analysis and results we had envisioned. Members of the team previously worked with the data visualisation software, Tableau. This was a brilliant asset to the team as it produced highly complex visualisations and dashboard outputs which became an integral part of the project. The team was very open and helpful in sharing each other's knowledge in different areas, such as engineering for example, and as a result it facilitated learning among the group - making us more well-rounded individuals in these technical areas as a result.

The dataset itself was a challenge at the beginning, as selecting an up to date one was hard to identify as well as having enough attributes to conduct analysis and KPIs. If we had more time for the project, we would have scraped up to date data from Daft.ie in order for our results to be more accurate while also potentially merging it with other data sources such as data from the Census of Ireland National Archives to include demographic information of the rental properties in an area.

Section 11 - Future Work

As mentioned in section 10 there were various challenges that the team encountered as well as lessons learned throughout the project. In reflection of these we were able to identify several aspects of our project that we could improve upon, if we were given the opportunity to carry out more research and follow different methods within the project.

Dataset: There are a few aspects of the dataset used that we'd like to improve upon in the future. We would aim to gather a larger sample size of properties around Dublin and the surrounding area. We would try to include historical data of rental properties, preferably between the last 5-10 years. This would allow us to work out the trend of prices from year on

year, potentially allowing us to predict prices into the future and not just at a moment in time. Another aspect we considered was to include datasets that have affected property prices and inflation rates. These could include economic, industrial or crime datasets. Although this may be challenging to score the attributes of these datasets in a way that could be applied to geographic locations in the model.

Other aspects we considered were to include demographic and geographic data of the regions in Dublin. By leveraging these datasets we could identify trends in rental prices based on the population's background and proximity of specific amenities. Another use if these datasets would allow users to filter their search based on demographic backgrounds of an area's populace or pick a property based on specific amenities.

By expanding the current dataset and by incorporating the dataset categories mentioned, it would facilitate the model to predict a more accurate rental price and possibly give us an insight into what aspects truly affect rental price.

Predictive model: An approach we would be interested in pursuing would be to incorporate linear regression components into a predictive model to facilitate a more accurate output. Regression analysis consists of mathematical methods which computes a continuous variable based on one or more variables [23]. In our case the rental price would be the continuous variable and this would be computed based on a select few variables, which would be property specifications, geographic location, proximity of amenities etc.

In order to pursue this method we would use Python programming to develop an algorithm to produce a predicted price. It would involve splitting the dataset into two separate datasets; a test dataset and a train dataset. The predictive model would be able to gain an understanding of the patterns from the training dataset, that could then be applied to the test dataset.

The performance of the predictive model would then be evaluated on two measures, bias and variance. Bias is based on the predictive model's proclivity to regularly interpret the dataset in an incorrect way, either by not considering all the data that's available or having missing items throughout the dataset. Having a complete and consistent dataset is essential in order to avoid producing an algorithm with a high bias. Variance is the proportion at which the output of the algorithm changes when different training data is used. We would aim to avoid this by developing a generic model that isn't catered to a specific training dataset [23].

Multiple linear regression is the form of regression analysis that we would incorporate into our Python algorithm as this type of regression deals with multiple variables and one output variable [23].

Interface: When the variance and bias reach a satisfactory level, we would aim to produce an interactive GUI that the predictive model feeds into. Possible libraries that we could make use of, would be the Kivy and Kivymd python libraries. By leveraging these libraries it would be possible to create an interactive interface that is adaptable to websites, android apps and IOS apps [24]. With the end goal being a user entering their own specifications that would predict the price of a property in a given area.

Although we may have been exposed to such methods and tools, we know it would take a considerable amount of time and expertise to design and implement a functioning and accurate application, from cradle to grave, that a user could interact with. It is our hope that we or another group of students, would pursue this idea in the future and bring this idea to market.

Walkthrough Video

YouTube Link: https://youtu.be/VlMM8EB4naM

Responsibility Statement

Team Member	Belbin Role	Responsibility	Group Marking
Aisling Keating	Specialist	Aisling was responsible for building the user interface in Tableau. In particular, she focused on the Data Analysis & Map tabs which allowed the user to explore the data.	Satisfactory
Anna Field	Teamworker	Anna along with Aisling was responsible for building the user interface in Tableau. In particular the Model tab which was used to predict the rental price.	Satisfactory
Colin Hehir	Implementer	Colin was responsible for sourcing the data from Kaggle, exploring & cleansing the data using Hive & Pig. He also assisted Niamh in sourcing the data.	Satisfactory
Ellen Jaye Woods	Plant	Ellen Jaye was responsible for managing the transmission of data between HDFS and Tableau. She experimented numerous different ways to achieve this as well as conducting a literature review.	Satisfactory
Niamh Eilis Feeney	Co-ordinator	Niamh was responsible for coordinating the team members, ensuring all tasks are progressing and deliverables are submitted on time. She was also responsible for recording the walkthrough video.	Satisfactory
Thomas Feeney	Complete Finisher	Thomas was responsible for scrutinizing our work and ensuring the highest standards. He also finalised the report ensuring inputs from everyone were synced and unified together, as well as exploring potential future work in this field.	Satisfactory

References

[1] Pope, C., 2021. House prices could rise by over 12% by year-end, survey shows. [online] The Irish Times. Available at: https://www.irishtimes.com/news/ireland/irish-news/house-prices-could-rise-by-over-12-by-year-end-survey-shows-1.4665462 [Accessed 14 November 2021].

[2] Curran, S., 2021. 'Chronic and worsening' supply crisis heaps pressure on renters as prices climb sharply. [online] TheJournal.ie. Available at: https://www.thejournal.ie/daft-rental-report-summer-5518739-Aug2021/ [Accessed 14 November 2021].

[3] Lyons, R., 2021. [online] Ww1.daft.ie. Available at: https://ww1.daft.ie/report/2021-Q3-houseprice-daftreport.pdf?d_rd=1 [Accessed 14 November 2021].

[4] Hennessy, M., 2021. Explainer: What are cuckoo funds and why are people complaining about them?. [online] TheJournal.ie. Available at: https://www.thejournal.ie/cuckoo-funds-explainer-4640142-May2019/ [Accessed 14 November 2021].

[5] moneyguideireland. 2021. Stamp Duty Rates in Ireland. [online] Available at: https://www.moneyguideireland.com/stamp-duty-rates-in-ireland.html [Accessed 14 November 2021].

[6] Finn, C., 2021. Dáil passes controversial amendment allowing cuckoo funds to avoid 10% stamp duty. [online] TheJournal.ie. Available at: https://www.thejournal.ie/10-stamp-duty-cuckoo-funds-5488173-Jul2021/?utm_source=story [Accessed 14 November 2021].

[7] Hehir, C. (2021). CA675 Cloud Technologies Assignment 1 Report.

[8] Kumar, M. (2016). Google cloud platform: a powerful big data analytics cloud platform. Int J Res Appl Sci Eng Technol, 4(11), 387-392.

[8] Nandimath, J., Banerjee, E., Patil, A., Kakade, P., Vaidya, S., & Chaturvedi, D. (2013, August). Big data analysis using Apache Hadoop. In 2013 IEEE 14th International Conference on Information Reuse & Integration (IRI) (pp. 700-703). IEEE.

[9] Fuad, A., Erwin, A., & Ipung, H. P. (2014, September). Processing performance on apache pig, apache hive and MySQL cluster. In Proceedings of International Conference on Information, Communication Technology and System (ICTS) 2014 (pp. 297-302). IEEE.

[10] Kumar, R., Gupta, N., Charu, S., Bansal, S., & Yadav, K. (2014). Comparison of SQL with HiveQL. International Journal for Research in Technological Studies, 1(9), 2348-1439.

[11] Gupta, Y.K. and Sharma, S., 2019. Impact of Big Data to Analyze Stock Exchange Data Using Apache PIG. International Journal of Innovative Technology and Exploring Engineering, 8(7), pp.1428-1433.

[12] Hive, A. (2013). Apache Hive.

[13] Kesavulu, M (2021). LOOP DCU: CA675 Cloud Technologies. [online] Available at: https://loop.dcu.ie/course/view.php?id=56668.

[14] IntelligentHQ (2021). Effective Use of Cloud Computing in Education. [image] https://www.intelligenthq.com/effective-use-cloud-computing-education/ (Accessed: 07 November 2021).

[15] Google Cloud (2021). Cloud Computing Services. [image] https://cloud.google.com/ (Accessed: 11 November 2021).

[16] Apache Hadoop (2021). Open-Source Software for Reliable, Scalable, Distributed Computing. [image] https://hadoop.apache.org/ (Accessed: 14 November 2021).

[17] Apache Pig (2021). Platform for Analyzing Large Data Sets [image] https://pig.apache.org/ (Accessed: 14 November 2021).

[18] Apache Hive (2021). Data Warehouse Software [image] https://hive.apache.org/ (Accessed: 14 November 2021).

[19] Tableau (2021). Data Visualisation Software [image] https://www.tableau.com/ (Accessed: 14 November 2021).

[20] Dwyer, H. (2021). JLL Bi-Annual Dublin Residential Forecasts Report for H2 2021. [online] https://www.jll.ie/en/trends-and-insights/research/jll-dublin-residential-forecasts (Accessed: 14 November 2021).

[21] Statista Research Department. (2021). Forecast of the percentage change on the previous year of residential property prices in Ireland from 2020 to 2022. [online] https://www.statista.com/statistics/1174799/residential-real-estate-price-forecast-change-in-ireland/ (Accessed: 14 November 2021).

[22] Lyu, G (2021). Dublin renting dataset scraped from Daft.ie(2020.9). [online] Available at: https://www.kaggle.com/d17129765/predicting-dublin-rental-daftie [Accessed 04 November 2021)].

[23] Kurama, V., 2019. Regression in Machine Learning: What it is and Examples of Different Models. [online] Built In. Available at: https://builtin.com/data-science/regression-machine-learning [Accessed 14 November 2021].

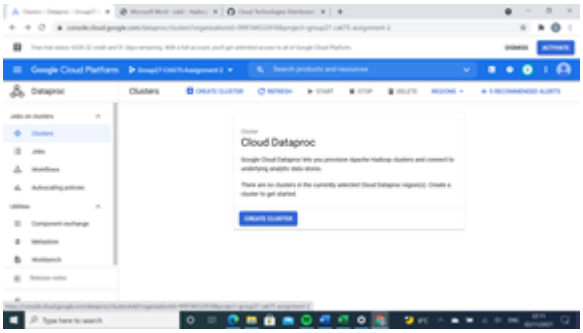
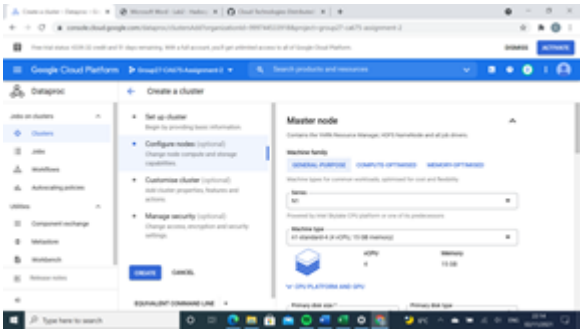
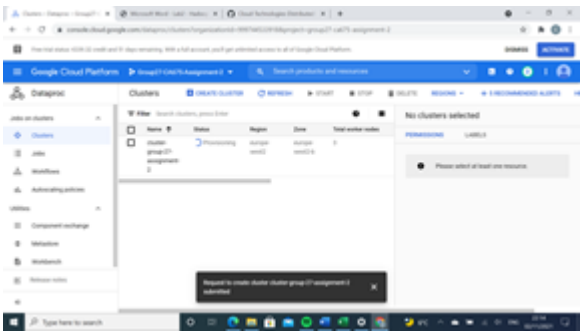
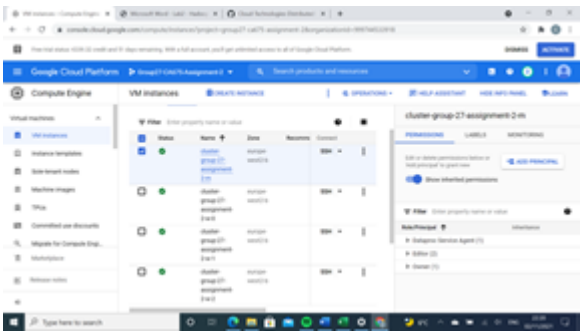
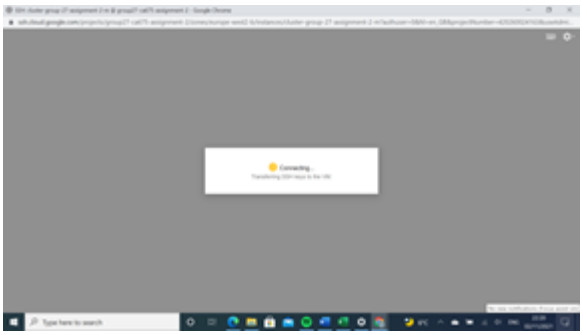
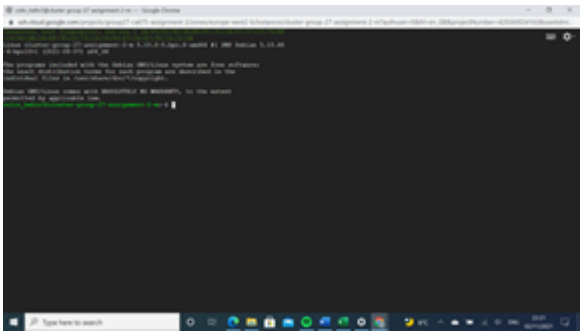
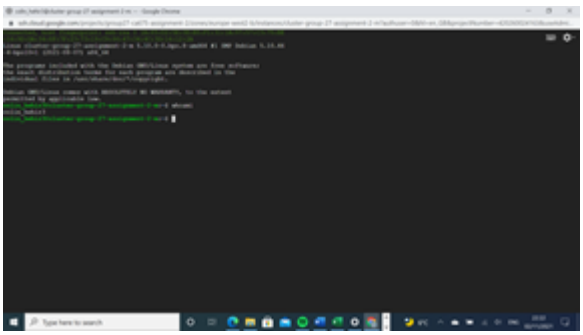
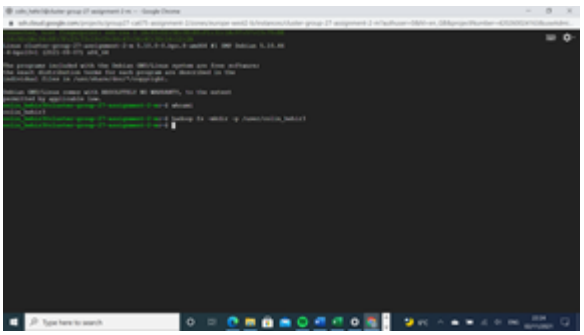
[24] Gupta, K., 2020. Deploying Machine Learning Models In Android Apps Using Python. [online] Analytics India Magazine. Available at:

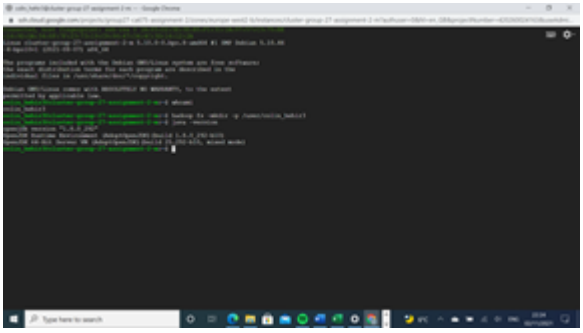
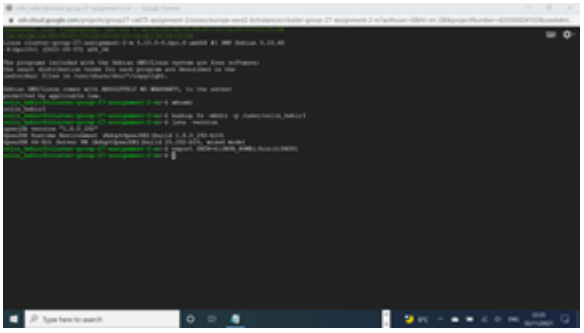
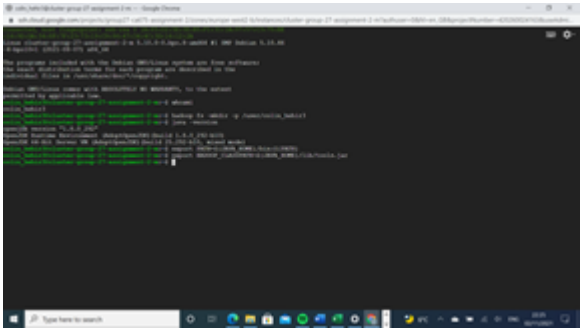
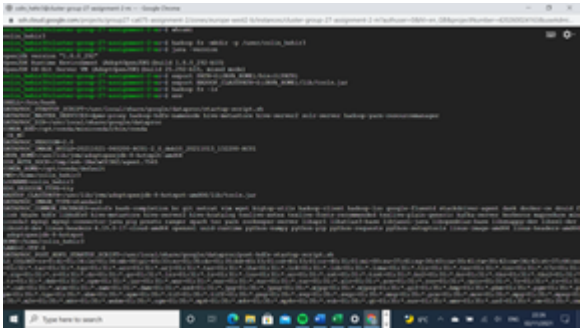
https://analyticsindiamag.com/deploying-machine-learning-models-in-android-apps-using-python/ [Accessed 14 November 2021].

Appendix

APPENDIX A: Environment Variables Set Up in GCP Hadoop Cluster

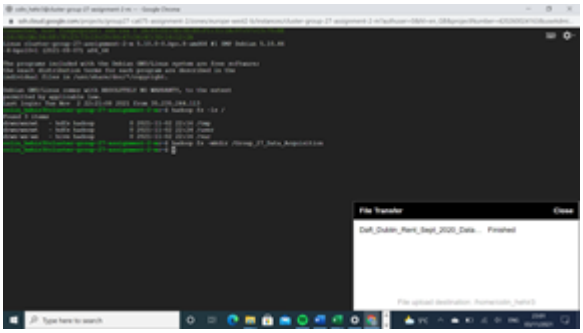
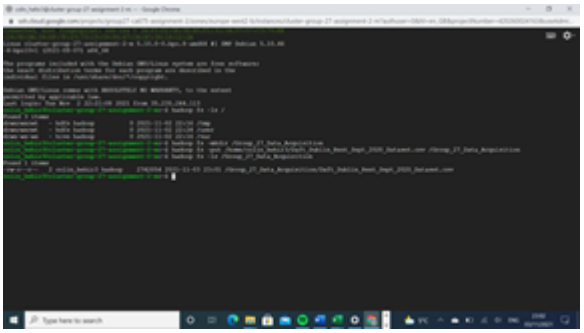
Please refer to the '1. Source Code - Environment Variables Set Up in GCP Hadoop Cluster.txt' file in the GitHub Repository for the relevant source code: colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)

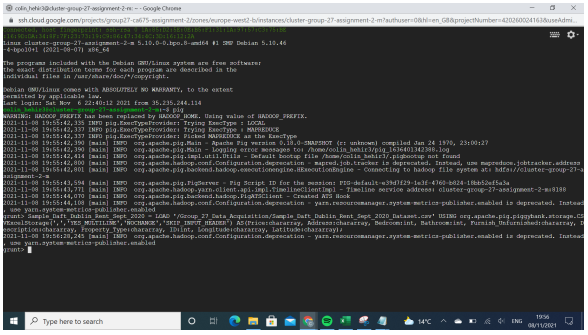
APPENDIX B: Load Data from Files into HDFS

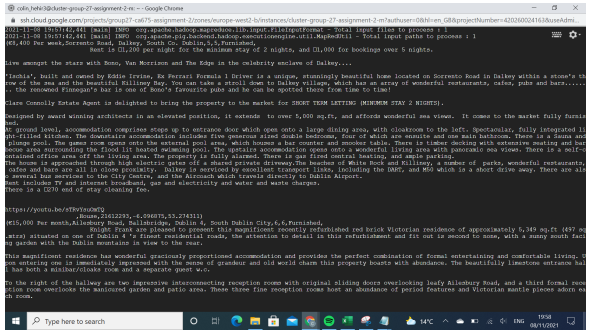
Please refer to the '2. Source Code - Load Data from Files into HDFS.txt' file in the GitHub Repository for the relevant source code:
colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)

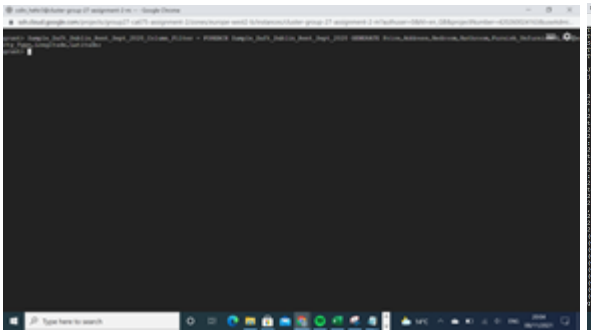



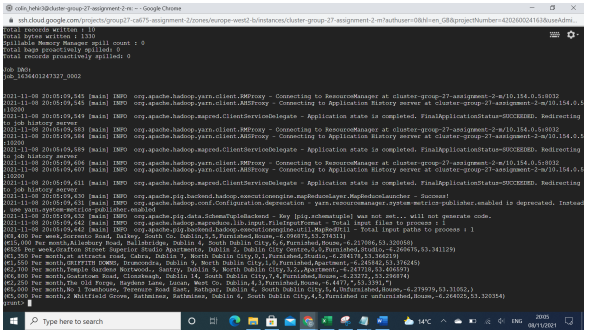
APPENDIX C: Pig ETL - Load Dataset and Column Filter

Please refer to the '3. Source Code - Pig ETL - Load Dataset and Column Filter.txt' file in the GitHub Repository for the relevant source code:
colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)
* Note the data in the screenshots only includes a sample of 10 from the dataset - in order for the viewer to efficiently see the respective data cleaning and processing taking place.





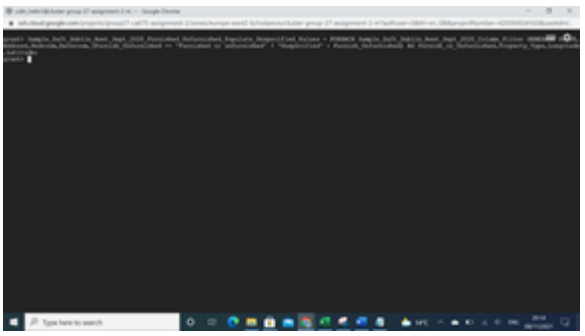


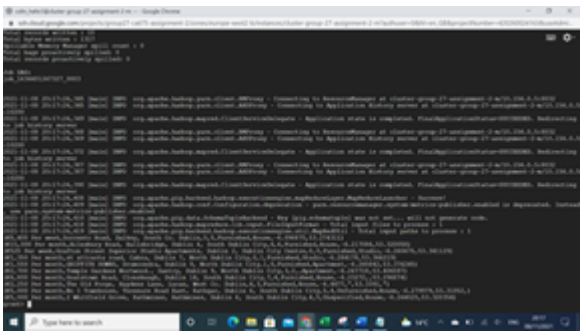


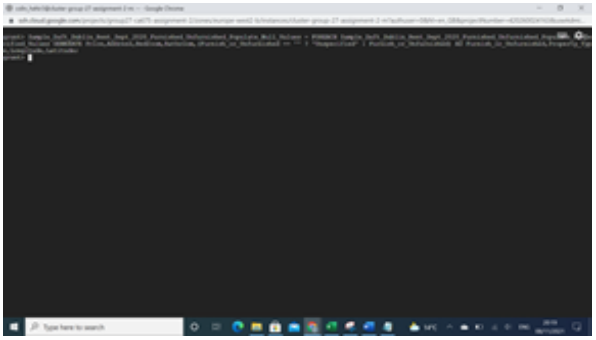
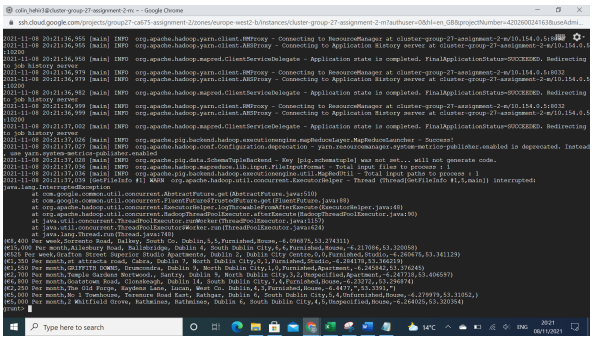
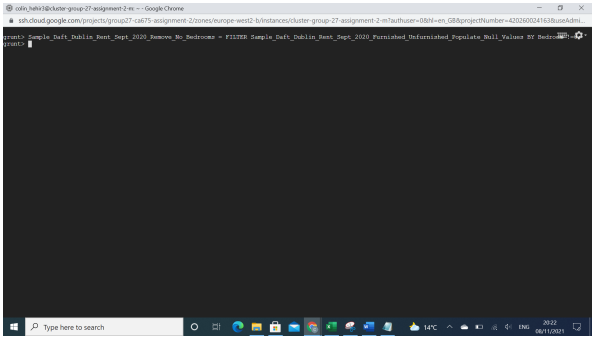
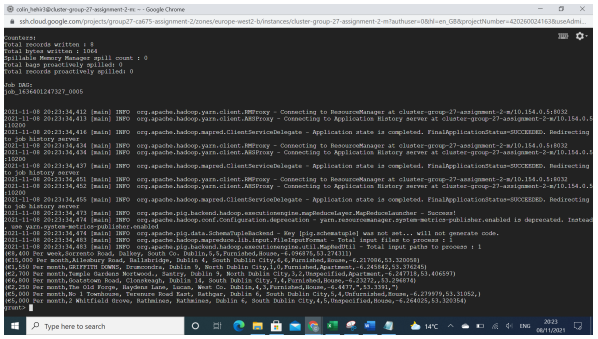
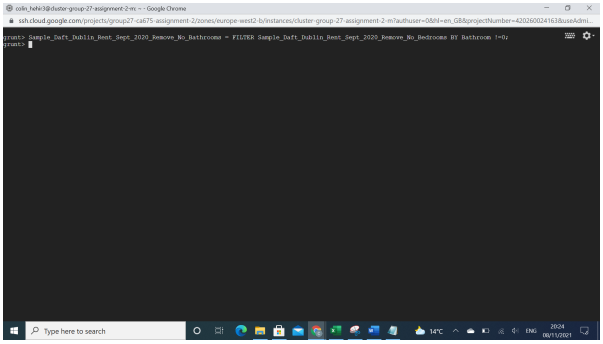
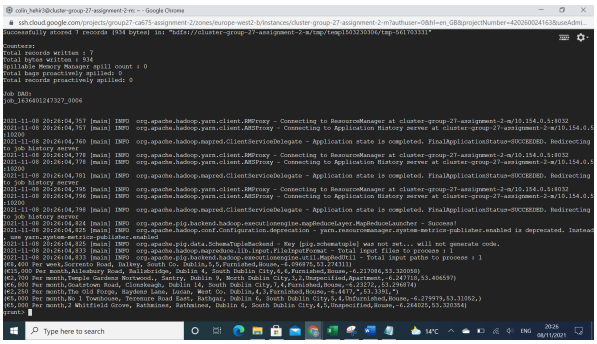
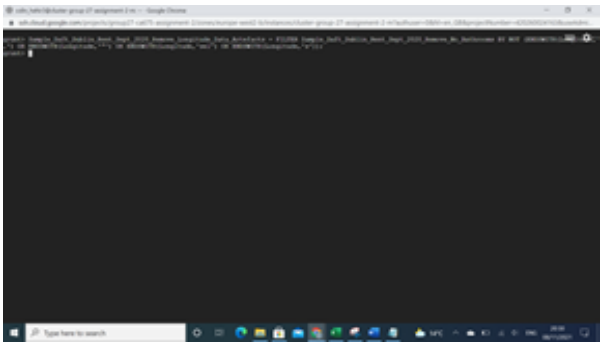
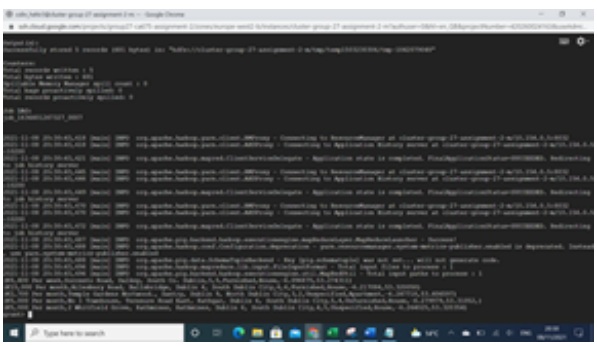
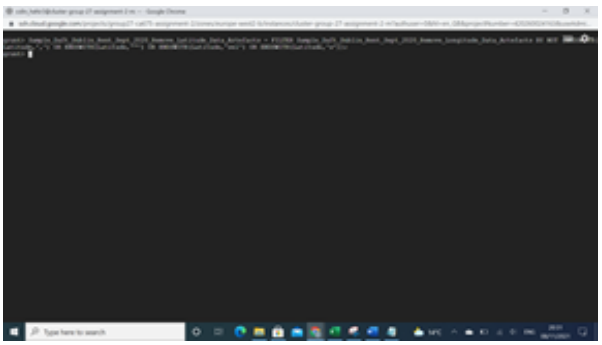
APPENDIX D: Pig ETL - Data Cleaning - Removing Data Artefacts and Replacing Null Values

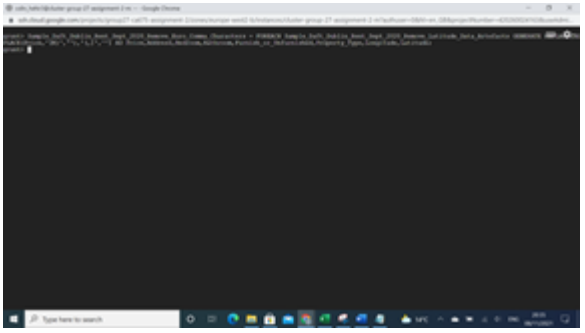
Please refer to the '4. Source Code - Pig ETL - Data Cleaning - Removing Data Artefacts and Replacing Null Values.txt' file in the GitHub Repository for the relevant source code: [colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)]

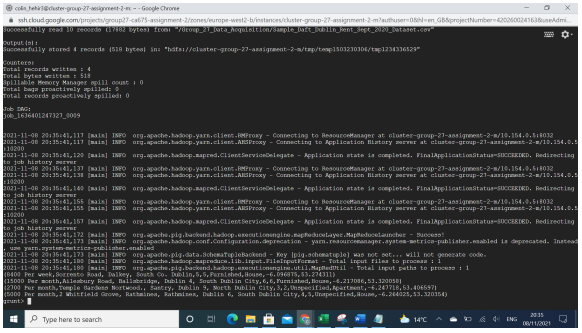
* Note the data in the screenshots only includes a sample of 10 from the dataset - in order for the viewer to efficiently see the respective data cleaning and processing taking place.





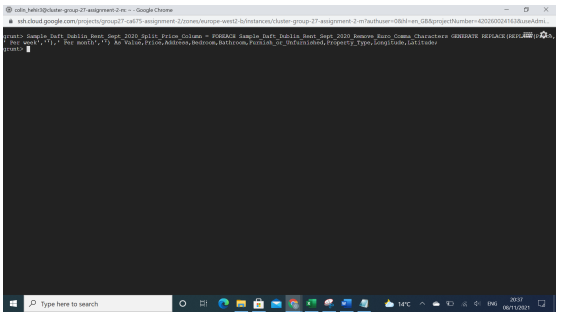


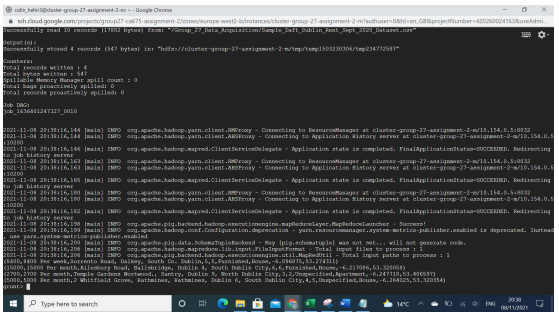


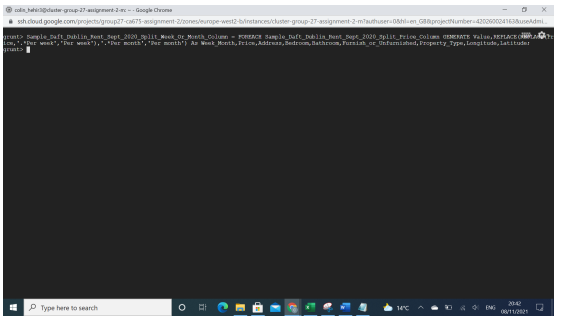
APPENDIX E: Pig ETL - Data Processing - Splitting Columns and Attributes

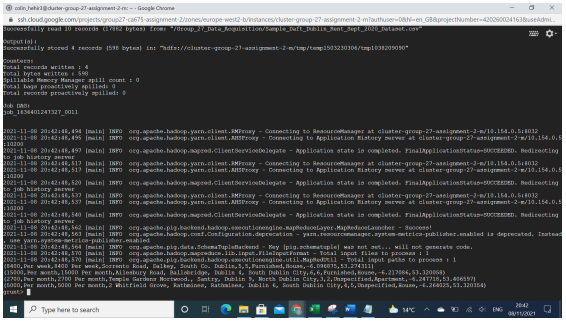
Please refer to the '5. Source Code - Pig ETL - Data Processing - Splitting Columns and Attributes.txt' file in the GitHub Repository for the relevant source code: [colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)]

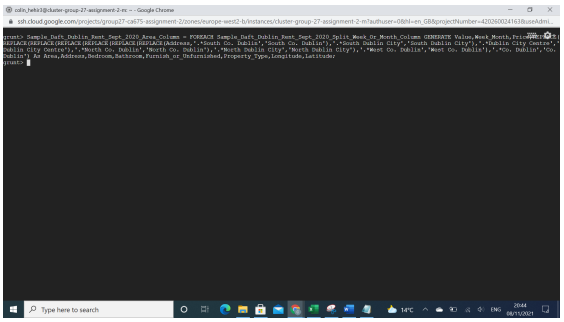
* Note the data in the screenshots only includes a sample of 10 from the dataset - in order for the viewer to efficiently see the respective data cleaning and processing taking place.

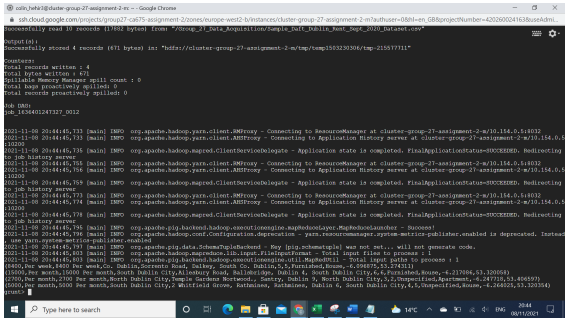


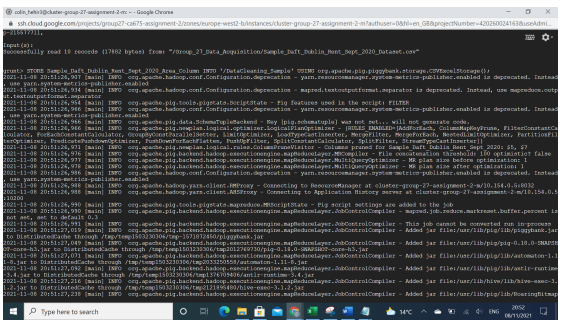


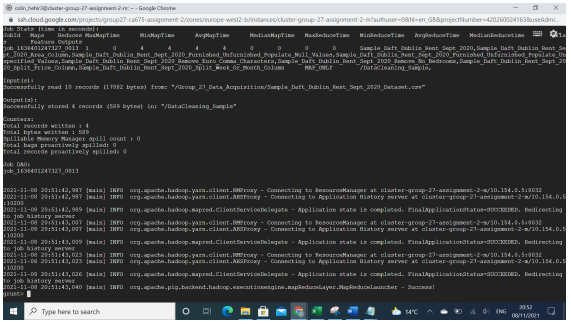






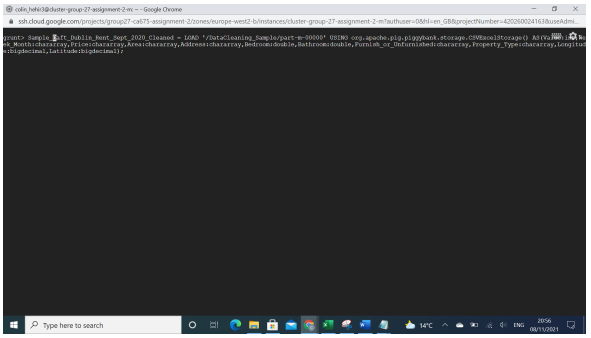


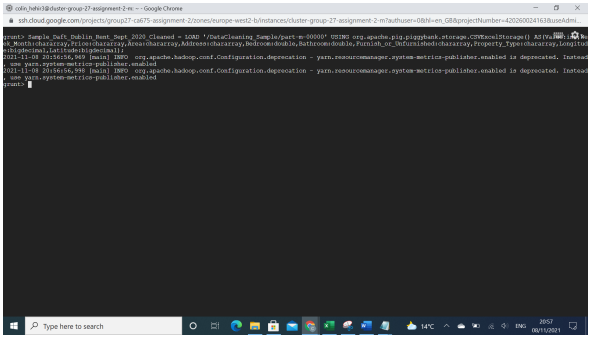


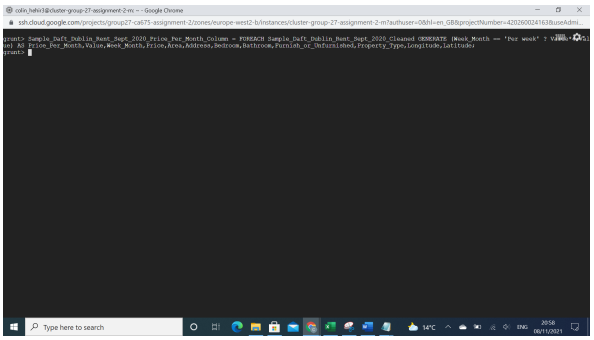


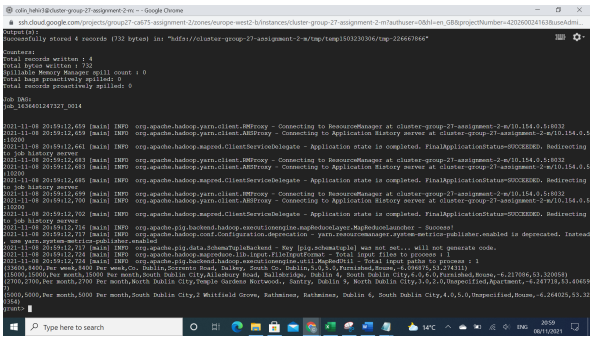
APPENDIX F: Pig ETL - Data Creation - Calculations and New Attributes

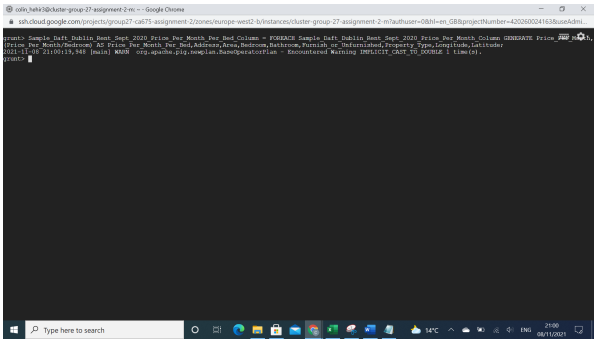
Please refer to the '6. Source Code - Pig ETL - Data Creation - Calculations and New Attributes.txt' file in the GitHub Repository for the relevant source code:
colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)
* Note the data in the screenshots only includes a sample of 10 from the dataset - in order for the viewer to efficiently see the respective data cleaning and processing taking place.

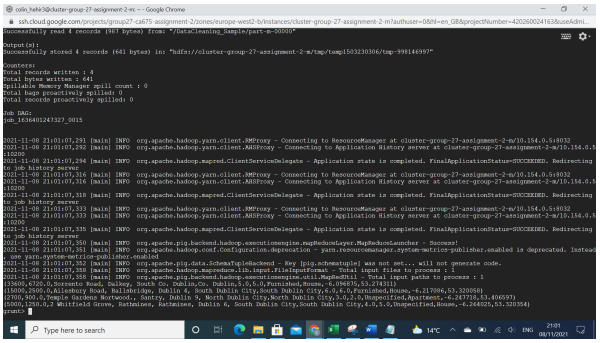


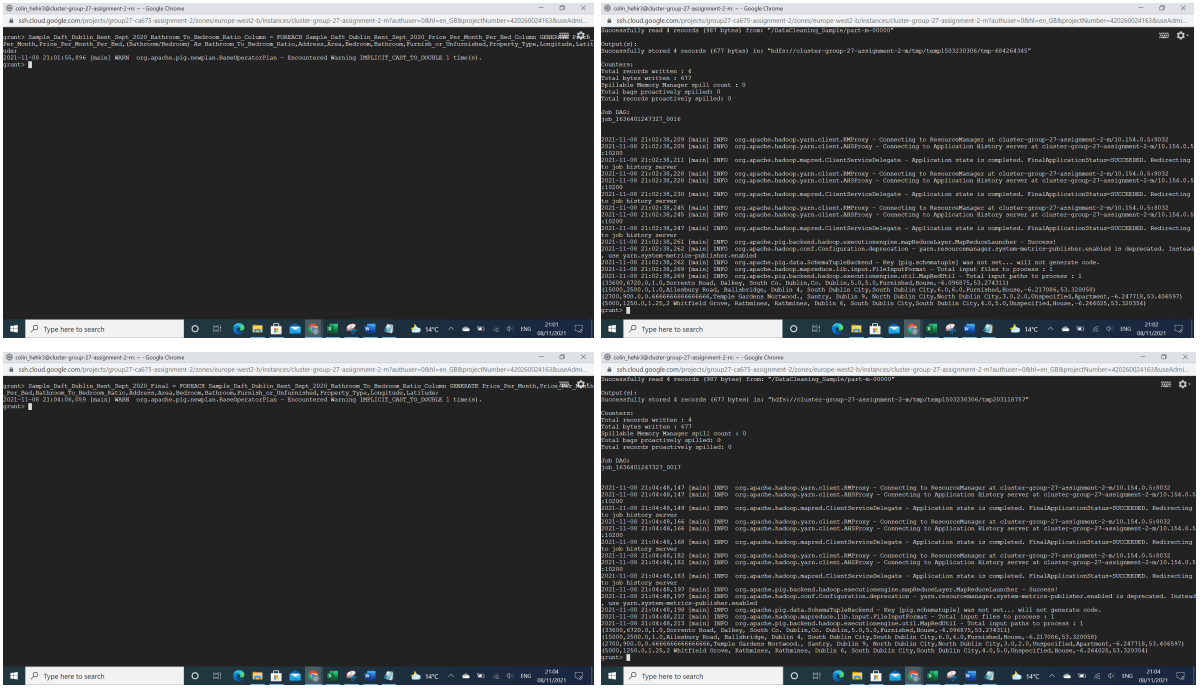






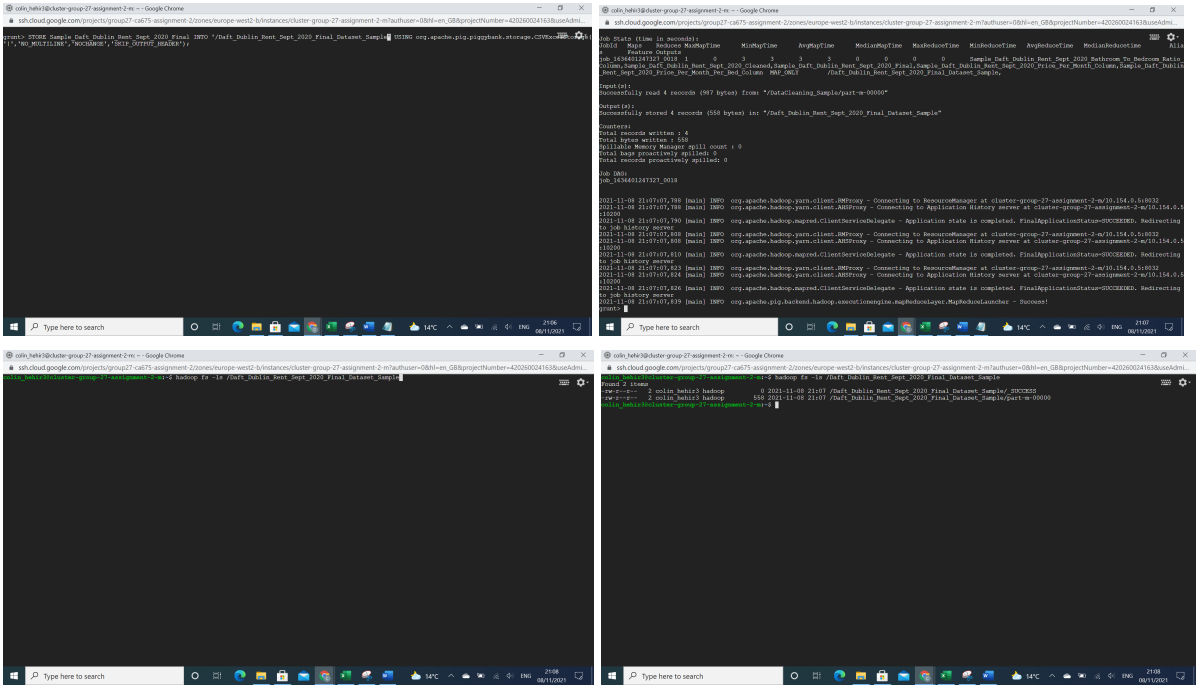






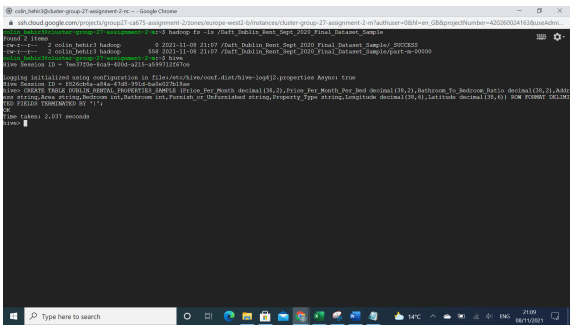
APPENDIX G: Pig ETL - Store New Dataset

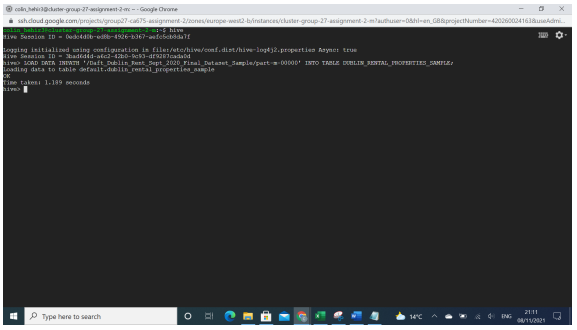
Please refer to the '7. Source Code - Pig ETL - Store New Dataset.txt' file in the GitHub Repository for the relevant source code:
colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)
* Note the data in the screenshots only includes a sample of 10 from the dataset - in order for the viewer to efficiently see the respective data cleaning and processing taking place.



APPENDIX H: Hive - Table Creation and Data Load

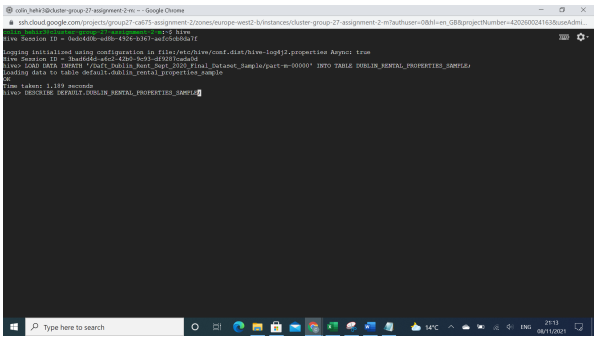
Please refer to the '8. Source Code - Hive - Table Creation and Data Load.txt' file in the GitHub Repository for the relevant source code:
colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)
* Note the data in the screenshots only includes a sample of 10 from the dataset - in order for the viewer to efficiently see the respective data cleaning and processing taking place.

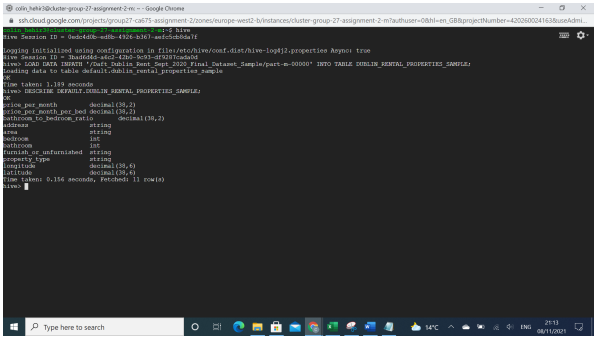


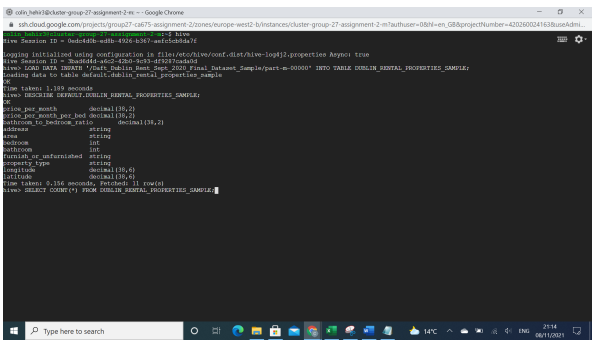


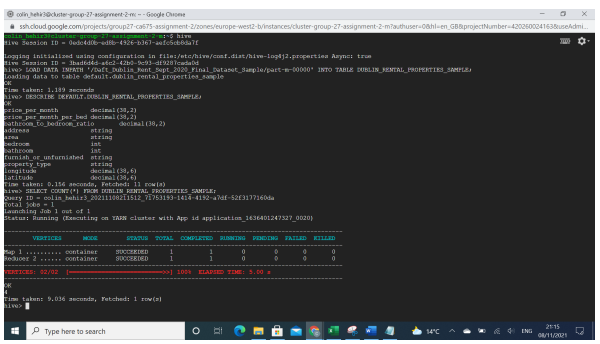
APPENDIX I: Hive Queries - Completeness and Accuracy Verification and Data Validation

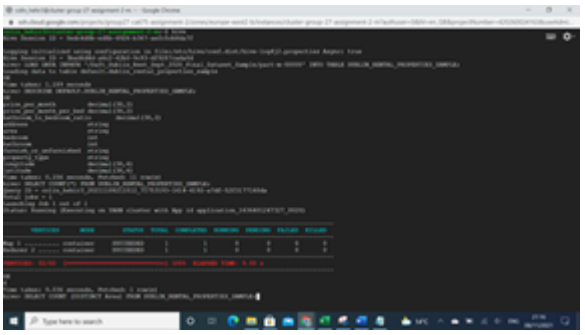
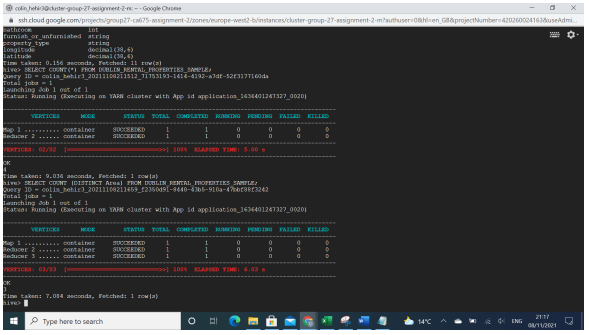
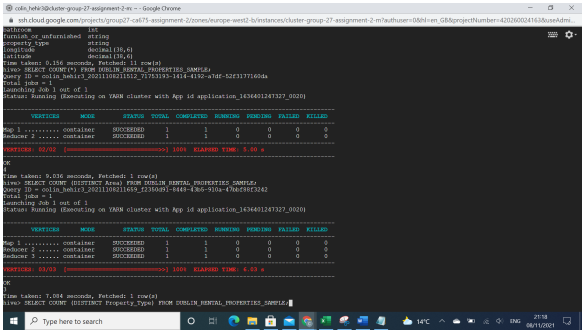
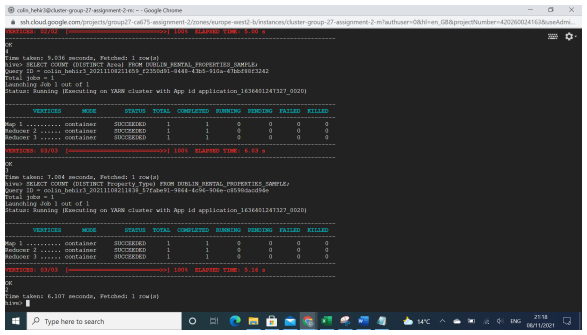
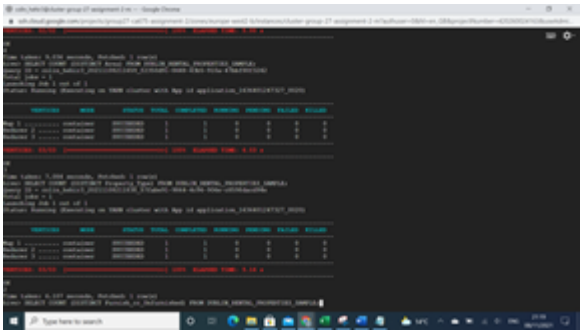
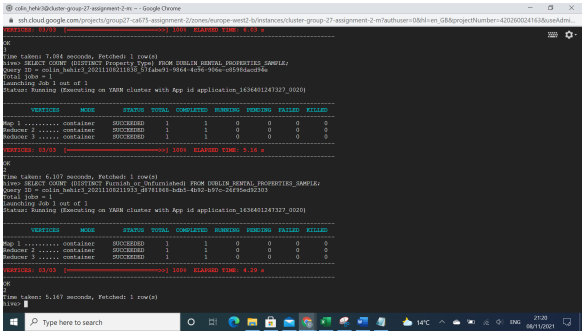
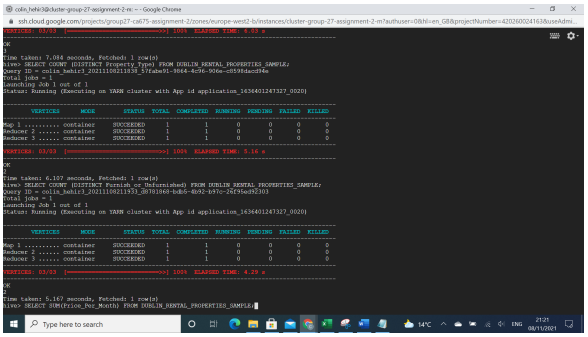
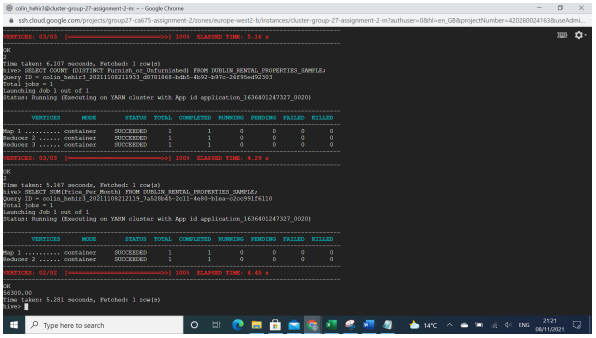
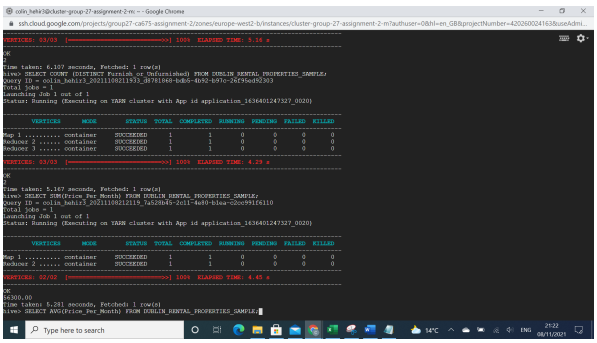
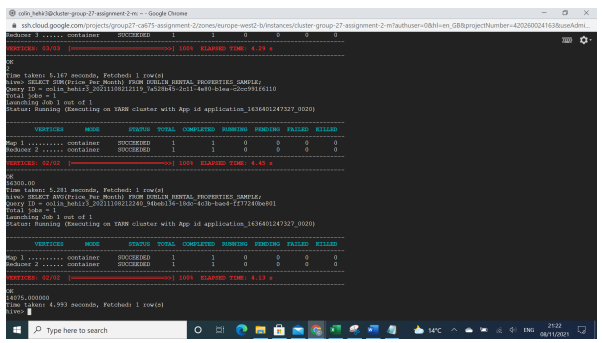
Please refer to the '9. Source Code - Hive Queries - Completeness and Accuracy Verification and Data Validation.txt' file in the GitHub Repository for the relevant source code: colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)
* Note the data in the screenshots only includes a sample of 10 from the dataset - in order for the viewer to efficiently see the respective data cleaning and processing taking place.

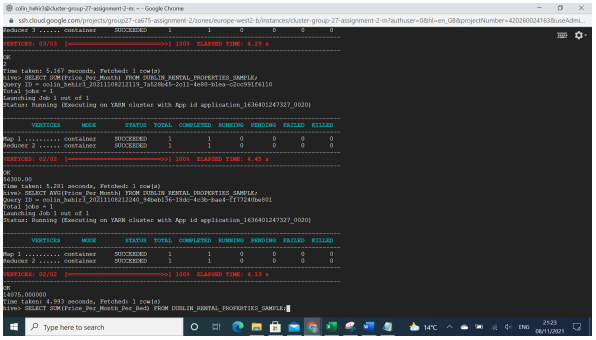
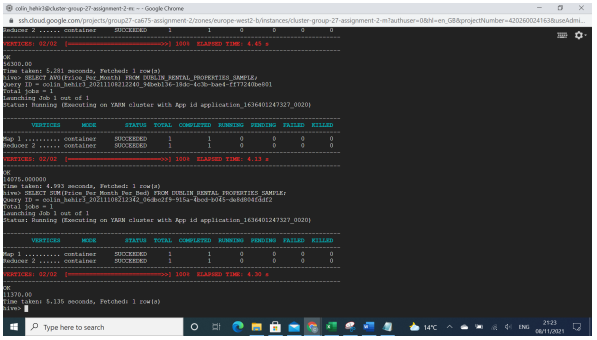
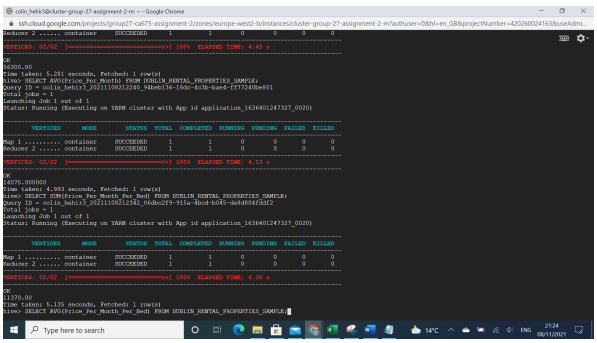
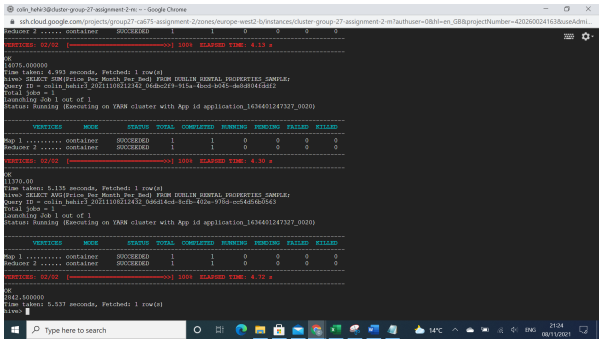
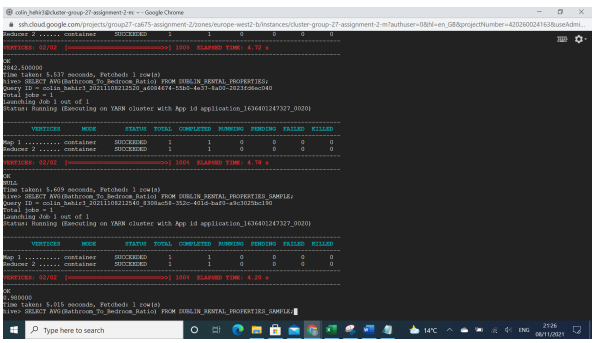
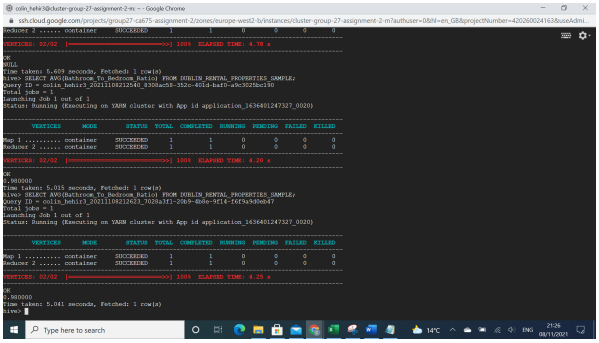








APPENDIX J: Data Storage and Source Creation

Please refer to the '10. Source Code - Data Storage and Source Creation.txt' file in the GitHub Repository for the relevant source code:
colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)

APPENDIX K: Data Model

Please refer to the '11. Source Code - Model via Apache Pig.txt' file in the GitHub Repository for the relevant source code:
colin-hehir/Cloud-Computing-User-Interface-Dublin-Rental-Properties-Analysis (github.com)