Cloud Computing

Assignment 2

Boston Housing Price Predictor

Anna Wisniewski (s3714689)

Mohammed Alotaibi (s3756190)

Contents

A. Contribution

Anna: 50%

Mohammed: 50%

B. Links

URL of the project: https://github.com/mohammedhalosaimi/Boston-Housing-Price-Predictor-App

The link for the web-application is not kept live due to costs associated with running the instances. If you would like to view the application, please deploy the application by following the Implementation section.

C. Summary

The purpose of this project was to design and create a practical application that utilises cloud platforms and technologies. In order to fulfil these requirements, a project that predicts the house prices in Boston's real estate market was developed. It is called the '*Boston Housing Price Predictor*'. The prediction was demonstrated by completing a multiple linear regression analysis on a dataset which contains a variety of features from Boston's real estate market. This data was loaded from the scikit-learn library [5]. The cloud technologies used in this project are primarily from AWS, specifically the Elastic Beanstalk, an EC2 instance, an S3 bucket [1] and the SNS service [6]. For the visualisation aspect of the project, Google's Data Studio was used [2]. The data for the visualisations was stored in Google's Big query [3]. Lastly, the website was developed using Flask with python [7].

D. Introduction

i. Motivations

Presently, easily accessible and mathematically sound real estate price prediction applications available for the general public to use do not exist. The purpose of the *Boston Housing Price Predictor* is to fill this void by giving people access to a regression-based application that will provide the user with accurate predictions about housing prices. An application like this can be used by the public to set realistic expectations about what they can and can't afford in Boston.

In competitive real estate markets, it is not uncommon for people to be told that a house is likely to be sold for a certain price at an auction (e.g. $1,000,000), despite the house being worth more [8]. Real estate agents occasionally partake in this legally grey area technique to encourage an audience at an auction. This disadvantages potential buyers because it tricks them into viewing a house as affordable. They may attend the auction, only to discover at the auction that the house is too expensive for them. In extreme cases, they may purchase the house despite it costing more than they expected, if they have already developed an attachment to the house. Situations such as these can cause great stress to potential home buyers. The *Boston Housing Price Predictor* will provide an additional source of information for those who are looking to buy property, so that they can have realistic expectations about what type of house they can afford.

ii. What does it do?

The *Boston Housing Price Predictor* will predict the median price of a house in Boston with a multiple linear regression. The values of the features used in the regression will be determined by the user's input in the web application. Once the user has selected a value for each feature, they can choose to have the results sent to their personal phone number if they wish. The user can also see a visual summary of the data with an interactive visualisation.

iii. Why is it required?

There are currently no applications or websites available that provide an unbiased prediction of house prices. This project has been created to show that it is achievable with a simple dataset, the Boston House Prices dataset. Therefore, the idea is applicable to real estate markets around the world. Particularly in places like Australia where the housing market is always a source of conversation and people are always looking for new ways to be informed prior to purchasing a house.

A lot of the predictions in the real estate market around the world tend to come from the media who have a vested interest in creating fear amongst the general public (i.e. 'House prices are predicted to hit an all-time low!' and 'Houses prices are so unaffordable, you will never be able to buy!'). This project will act as a prototype for an application that will provide an alternative to house price predictions from the media.

iv. How it can be used as real-life application

The *Boston Housing Price Predictor* can theoretically be at least partially replicated in any housing market around the world so that it can be used as a 'real-life' application. It can only be partially replicated because the developer will need to build the project on data from the real estate market that they would like to run the prediction on. Prior to replicating the project, the developer will need to dedicate time to data pre-processing and data exploration. This is so the correct machine learning technique can be chosen for the prediction aspect of the web-application.

However, the concept remains the same. Furthermore, the same software and architecture can be used to achieve the same result. The web-application can still be used to advise people of housing prices in any area that it is set up to complete this. The user can use it to influence their 'real-life' decisions on whether to purchase a house and how much for.

v. The advantages of this web-application

The main advantages of the *Boston Housing Price Predictor* are that it fills the void of machine-learning based applications for real estate markets and it is very simple to use. The tools that are currently available tend to be run by banks, media organisations or real estate advertising platforms. Therefore, it is likely that these businesses have their own agendas. Furthermore, they often contain a lot of information and jargon that can be difficult to sift through, particularly for first home buyers. Therefore, people currently rely on these difficult to interpret services or personal biases alone to make their decisions in the real estate market. Their opinions can be heavily swayed by a story they heard from a friend, or their singular negative experiences. *Boston Housing Price Predictor* is a tool, backed data science, that users can add to their tool belt to make good and informed decisions. When people have access to these tools, they are more likely to make their decisions based on real facts, instead of their own personal biases.

vi. Audience

The intended audience of the *Boston Housing Price Predictor* is people who are looking to purchase real estate in Boston who are unfamiliar with the real estate market and what influences housing prices. It was also be aimed at people who are looking for an alternative to the media scare campaigns on house prices.

E. Related Work

There are several online resources that have similar price predicting capabilities to the *Boston Housing Price Predictor*. The first three resources listed in this section are Australian resources, however there are likely to be equivalent resources available to Boston. They are still deemed as 'related work' for this project because the *Boston Housing Price Predictor* can be replicated to real estate markets around the world, as described in section D.iv 'How it can be used as a real-life application'. The related works are as follows:

1. https://www.domain.com.au/

Domain has a website and a mobile application. Their services are frequently utilised by Australians who are looking to rent or purchase a house. Each listing has a picture of the house, an estimate of its cost, a description, and information about potentially important information related to the house and its location (e.g. is the house in a school catchment zone). However, this cost is not set by any machine learning methods. It is set by the real estate agent [8],[9]. Therefore, these advertisements, albeit sometimes accurate and subject to certain laws, are prone to being influenced by real estate agents' own agendas.

2. https://propertyupdate.com.au/property-predictions-for-2021-revealed/#3

Articles, such as this one from propertyupdate.com, are frequent in the real estate market area. However, they are often filled with real-estate market specific jargon that can be difficult to navigate for people who have next to no knowledge of the real estate market. They also have a lot of information in them This is a great thing if the user wants all that information. However, if the user wants a quick snapshot of what they estimate the house price to be, these types of reports will not suit them. The user will need to sift through all of the information in the hope that they find something relevant to them. Lastly, even if the user finds something relevant to them, the information is unlikely to be targeted specifically to their needs.

3. https://www.anz.com.au/personal/home-loans/calculators-tools/property-profile-reports/

This ANZ service will, on request, send the user an automatically generated report on a particular property. This report summarises publicly available information about that property. ANZ advertises this service as 'Australia's most accurate free property price predictor'. The user will be required to fill out a form with their contact details and the address of the house that they would like the property report for. They receive the property report in their email inbox shortly after. It may very well be true, that this is Australia's most accurate and free property price predictor. However, it requires the user to have a specific house in mind. It also requires the user to read through a report, which they may not want to do.

4. https://www.kaggle.com/sagarnildass/predicting-boston-house-prices

The Boston housing dataset is famous in the machine learning community because it is a good dataset to demonstrate how a regression analysis works. A lot of analysis has been done on this dataset before and can be seen online on websites like Kaggle. The analysis on its own (e.g. on Jupyter notebook) is not useful to the target audience of the *Boston Housing Price Predictor*. However, the different types of analysis that people have completed was the inspiration behind the regression analysis used in this project's application.

F. Software Design/Architecture

i. Architectural diagram of the Boston Housing Price Predictor Application

The design of the cloud computing architecture used to create the *Boston Housing Price Predictor* can be seen below. It was designed as a distributed model to ensure that every required task has its own dedicated infrastructure. In other words, each task does not have to share its resources with another task.

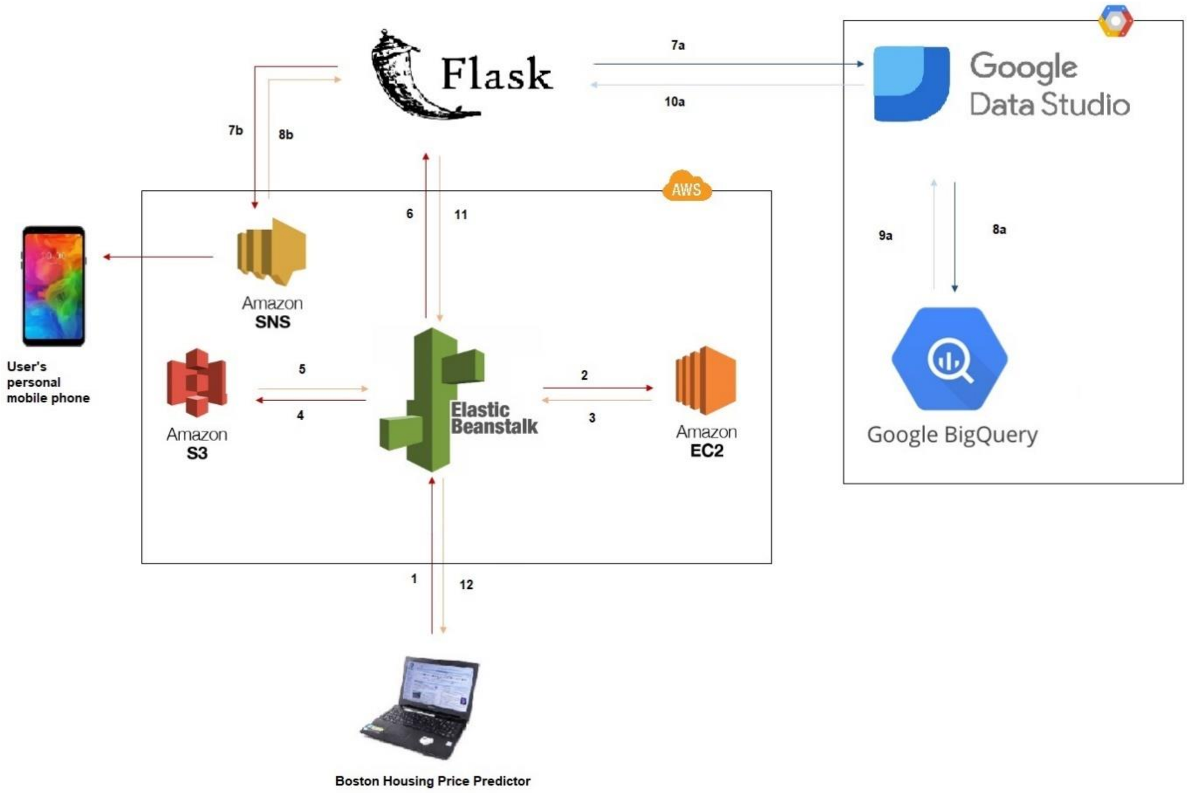


Figure 1. Boston Housing Price Predictor Architectural Diagram

Each number in Figure 1 corresponds to a communication task between the two infrastructure pieces on either side of the arrow.

The Amazon Elastic Beanstalk was chosen as the method to deploy the application because it is a relatively fast and simple tool to use. It automatically created the S3 bucket and the EC2 instance which are required for the deployment. The Amazon S3 bucket is where the source code of the Boston Housing Price Predictor is stored when the environment is created and accessed when the program is deployed. The Amazon EC2 instance is a virtual machine that is used to run the web-applications

Amazon's SNS is used to send the results of a user's regression analysis to their own mobile phone. This is so that the user does not have to remember any of the results they generated because the results are not stored anywhere.

Originally an attempt was made to generate the visualisations from Amazon's QuickSight. However, due to several issues with this infrastructure, it was decided that the visualisations would be created with Google Data Studio instead, from the Google Cloud Platform. The Google Cloud Data Studio allows an interactive visualisation to be present on the application so that users can get a visual summary of the data. This will help users who want to get a 'quick picture' of the current state of the market, based on the data, without having to go through the regression process. The data for the visualisations was stored in Google's Big Query. Lastly, the web-application was created using Flask.

A description of each communication task and the infrastructure piece is as follows:

1. The user accesses the web-application through the URL provided by the Elastic Beanstalk.

2, 3, 4, 5, 6. The Elastic Beanstalk runs the Flask application by using the S3 bucket and the EC2 instance.

7a. If the user clicks the button for the visualisations, the flask application will send the user to the Google Data Studio visualisation.

8a. The visualisation will load using the data from Big Query.

9a. The data from Big Query will be sent back to the Google Data Studio and be seen by the user.

7b. From the main page, if the user inputs the data, and enters their number to receive their results, the Flask application will trigger the SNS, and the user will receive their results.

8b, 11, 12. The Flask application will also know that the SNS service was used and will state this in a message on the results page. The user will also receive the result via the web-application.

ii. Description of the dataset

The Boston House Price Dataset is a famous dataset from scikit learn. The information in the dataset was put together by the US Census Service. These descriptions have come directly from scikit-learn. Each attribute is numerical, except for the CHAS attribute. A description of each of the attributes is as follows:

1. CRIM: per capita crime rate by town
2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
3. INDUS: proportion of non-retail business acres per town
4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
5. NOX: nitric oxides concentration (parts per 10 million)
6. RM: average number of rooms per dwelling
7. AGE: proportion of owner-occupied units built prior to 1940
8. DIS: weighted distances to five Boston employment centres
9. RAD: index of accessibility to radial highways
10. TAX: full-value property-tax rate per $10,000
11. PTRATIO: pupil-teacher ratio by town
12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
13. LSTAT: % lower status of the population
14. MEDV or PRICE: Median value of owner-occupied homes in $1000's

iii. Machine Learning Technique

The machine learning technique used to run the prediction in the *Boston Housing Price Predictor* is Multiple Linear Regression. This machine learning technique is used to model the relationship between a target variable and at least two features [4]. In this project, the target variable is MEDV (listed as PRICE in the Google Data Studio) because this is the attribute that we are predicting. All the features are used as the explanatory variables (i.e. the regression model will summarise how these features explain, or predict, the target variable). The model is created by fitting a linear equation to the dataset. This type of regression is only suitable for the dataset when the relationship between the features and the target variable is linear. Therefore, it is important to note that there must be a thorough analysis done on any dataset prior to implementing a multiple linear regression model to ensure that the results of the regression are an accurate prediction.

G. Implementation Guide

i. Technologies

Prior to following the implementation guide, the developer will need to ensure they have access to the following technologies:

1. Programming Languages
    a. Python (3). This project was completed with Python 3.7
    b. Jupyter Notebook
    c. Flask – Web Framework
2. Integrated development environment (IDE)
    a. Microsoft Visual Studio Code
3. Accounts
    a. AWS Account (This will project will not work with an Educate account)
    b. Google Account
4. Cloud Platforms
    a. AWS Elastic Beanstalk
    b. AWS S3
    c. AWS EC2 Instances
    d. Google Big Query
    e. Google Data Studio
    f. AWS SNS
5. Packages
    a. Matplotlib 3.1.1
    b. boto3 1.9.243
    c. seaborn 0.9.0
    d. pandas 0.25.1
    e. numpy 1.17.2
    f. scikit_learn 0.21.3

ii. Jupyter Notebook

Jupyter Notebook was used to access the dataset, pre-process the data and do the multiple linear regression analysis. The full analysis for this project can be seen on the github link provided, under the file name *Boston house pricing.ipynb*. This file can be used to replicate the project. The developer will need to do the following steps:

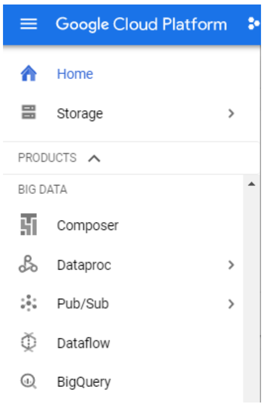
1. Import the packages
2. Import and load the dataset using sklearn 'from sklearn.datasets import load_boston'
3. Explore the data (i.e. check its shape, descriptions, etc…)
4. Convert the dataset into a pandas data frame. This is so it can be easily manipulated for the analysis.
5. Assign feature names to the columns
6. Add the PRICE column (target variable) to the data frame
7. Check the correlations between the features by creating a heatmap
8. Split the data into the features and the target variable.
9. Split the data into training and test sets
10. Standardize the data, so that it is suitable for the analysis.
11. Instantiate the classifier
12. Train the model
13. Print the results
14. Create a separate data frame for the prediction from the training dataset

Jupyter Notebook is used for the initial analysis because it is a great IDE to perform machine learning problems on. Once this process is complete, the developer can move onto the other aspects of the application, before eventually putting it all together on the Flask application.

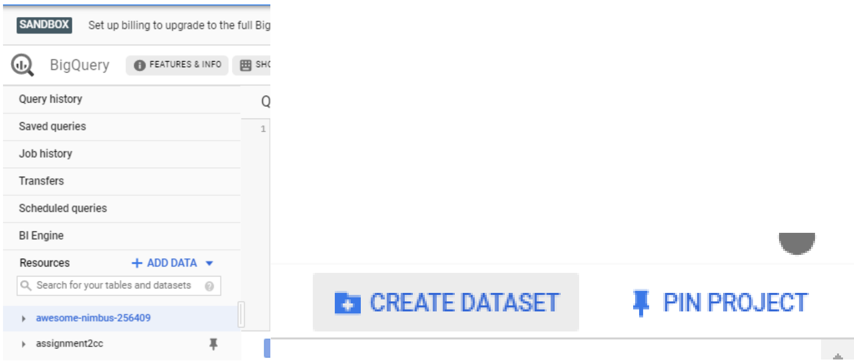
iii. Google Data Studio

Google Data Studio is used to visualise the data in the Boston Housing Price Predictor Application. As all the features but one (CHAS) in the Boston dataset are numerical, making meaningful visualisations can be difficult. They are also not that appealing to viewers. Therefore, prior to uploading the dataset to the Google Data Studio, a separate Jupyter Notebook is used to create categorical variables of each attribute. To see how this is done, open the notebook called *bostoncsv_anna.ipynb* from the github link. In this notebook, a visualisation has been made for each feature so that the developer can easily see which categorical variables would make the most sense to create. They were all created for the purposes of this project. When this preprocessing is complete, the Boston dataset can be exported as an excel file (last line in the Jupyter Notebook) and uploaded to BigQuery. Follow the following steps in order to upload the dataset to BigQuery and create the visualisations:

1. Open BigQuery from the Google Cloud Platform: https://cloud.google.com/

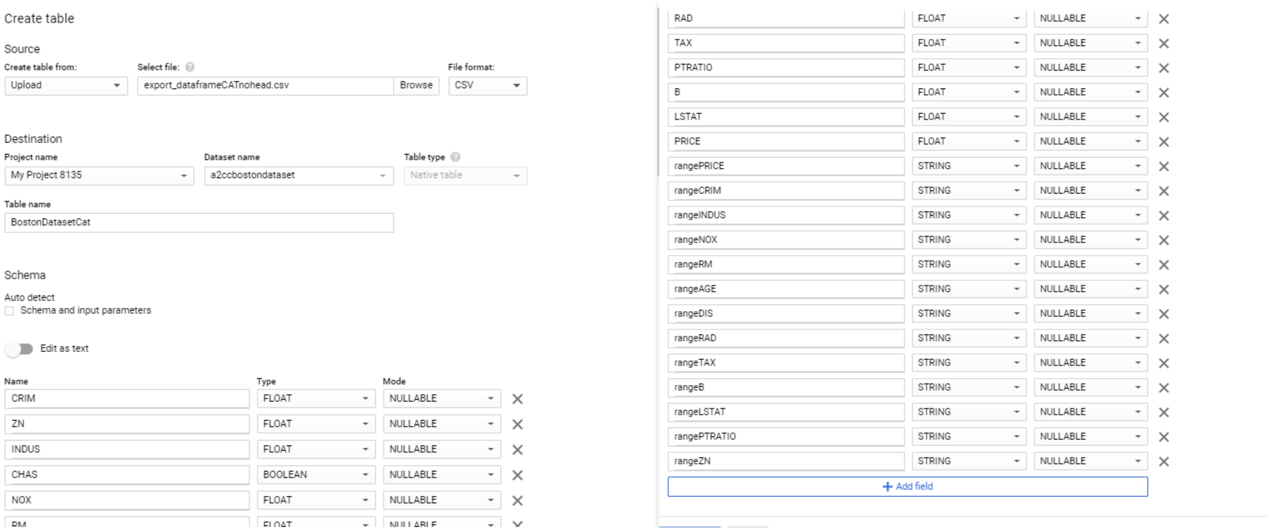


2. Click on the project that you would like to place the data in. Next, click on 'CREATE DATASET' in the bottom right hand corner.



3. Choose a Dataset ID and then click 'CREATE DATASET'.

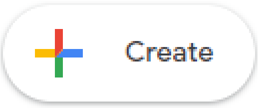
4. Open the new dataset that you created and click 'CREATE TABLE'. Populate the fields like the image below. All the original variables are the FLOAT type, except for CHAS which is Boolean. Furthermore, all the variables which start with range in their name are the STRING type. Click 'CREATE TABLE' when this is complete

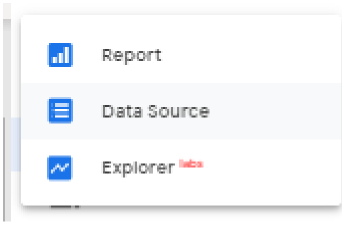


5. The Table will be visible if the data was imported successfully.

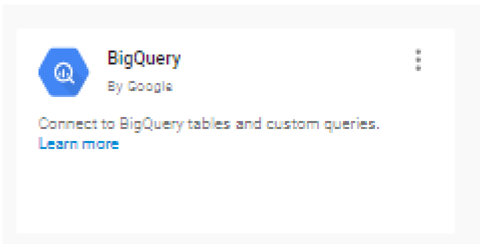
Next, the user will need to login to their google account from the Google Data Studio page: https://marketingplatform.google.com/about/data-studio/. Follow the following steps to create a visualisation that can be embedded in the web-application:

6. Click the Create button on the left-hand side, and then click 'Data Source':

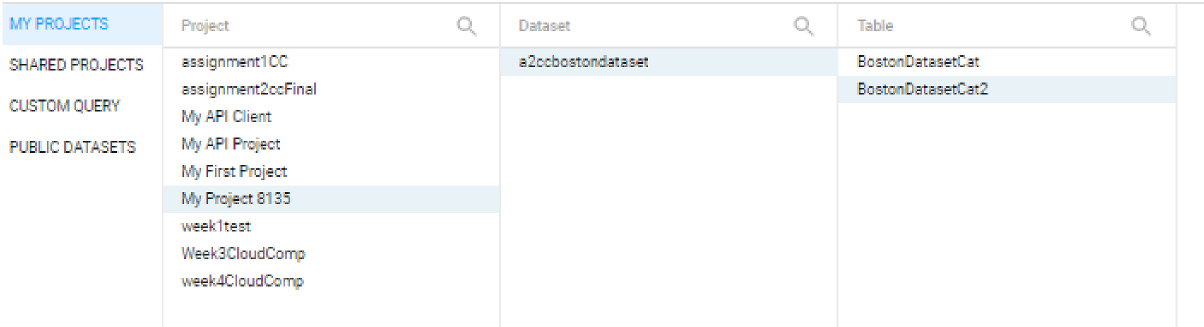




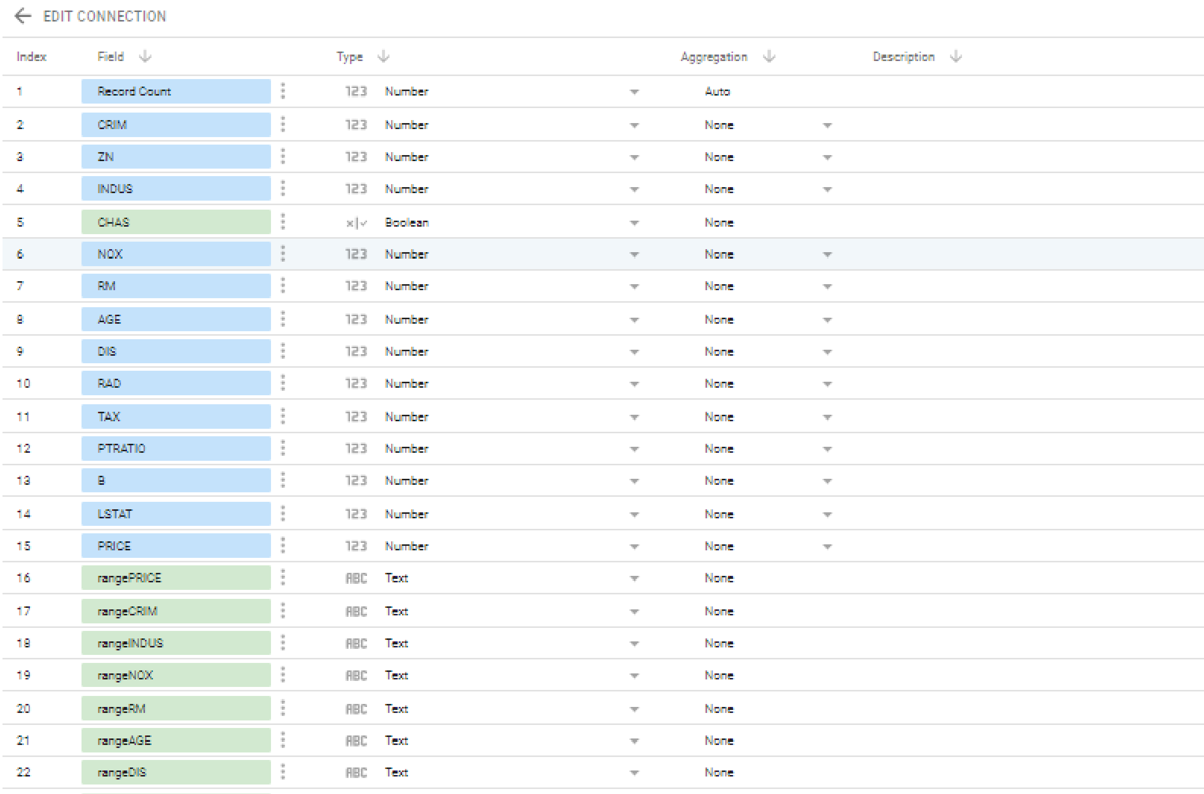
7. Click 'BigQuery'



8. Select the project where you placed the dataset, dataset and table where you placed the dataset. Then click 'Connect' in the top right-hand corner.

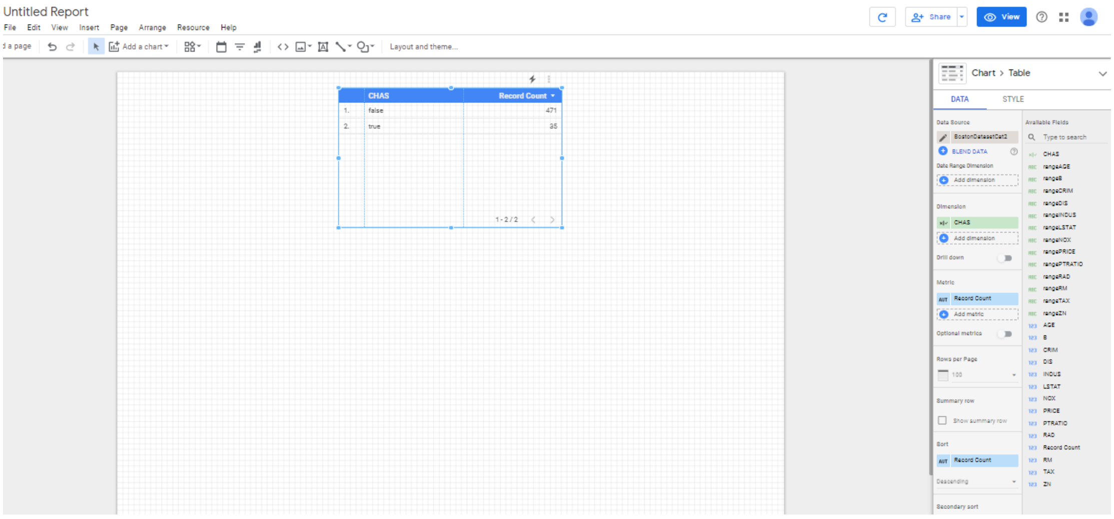


9. Have a scan through the data and make sure every attribute is the right type. You can change the 'Aggregation' of a variable here if you wish.

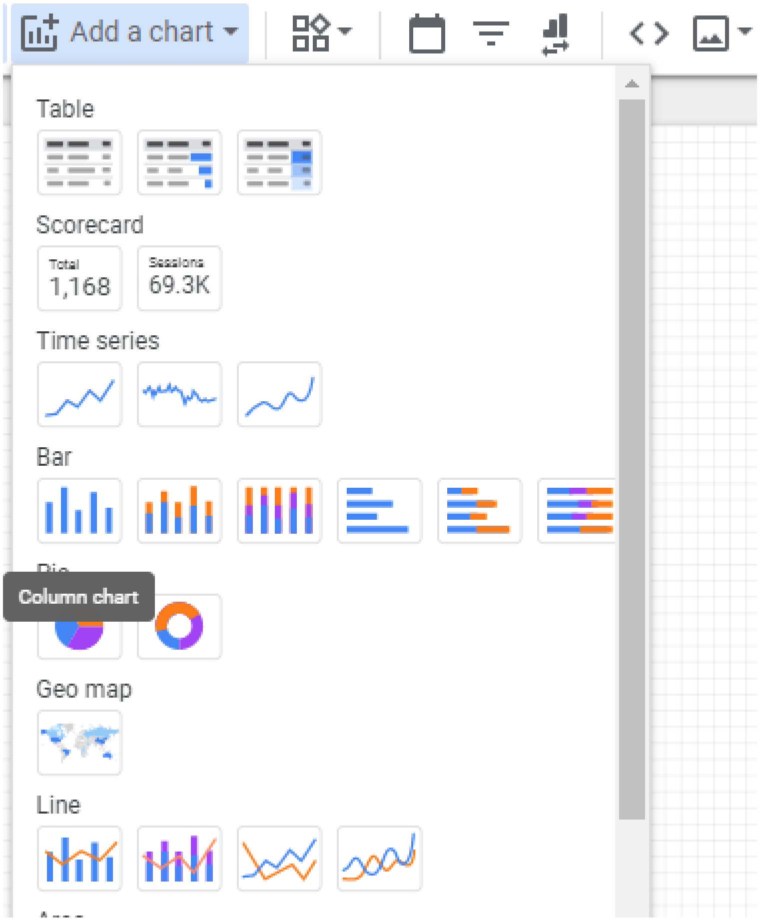

EDIT CONNECTION

Index	Field	Type	Aggregation	Description
1	Record Count	123 Number	Auto	
2	CRIM	123 Number	None	
3	ZN	123 Number	None	
4	INDUS	123 Number	None	
5	CHAS	×|✓ Boolean	None	
6	NOX	123 Number	None	
7	RM	123 Number	None	
8	AGE	123 Number	None	
9	DIS	123 Number	None	
10	RAD	123 Number	None	
11	TAX	123 Number	None	
12	PTRATIO	123 Number	None	
13	B	123 Number	None	
14	LSTAT	123 Number	None	
15	PRICE	123 Number	None	
16	rangePRICE	ABC Text	None	
17	rangeCRIM	ABC Text	None	
18	rangeINDUS	ABC Text	None	
19	rangeNOX	ABC Text	None	
20	rangeRM	ABC Text	None	
21	rangeAGE	ABC Text	None	
22	rangeDIS	ABC Text	None	

10. Click 'Create Report' in the top right-hand corner when you are ready to start making the dataset. Click 'Add to Report' when the pop askes you again.
11. Once the data has been added to the report successfully, you will see a screen like the one below. This is where you can make the visualisations that will be embedded in the web application. The table has been automatically generated as the first visualisation.

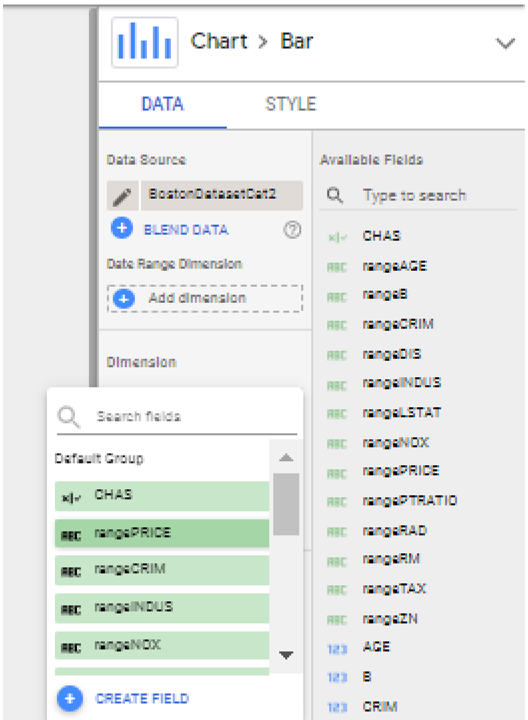


12. Click the table in the middle of the report and delete it. In the top banner, click ‘Add a Chart’. There is a list of all the different charts that you can select. Choose the first ‘Bar’ chart type.

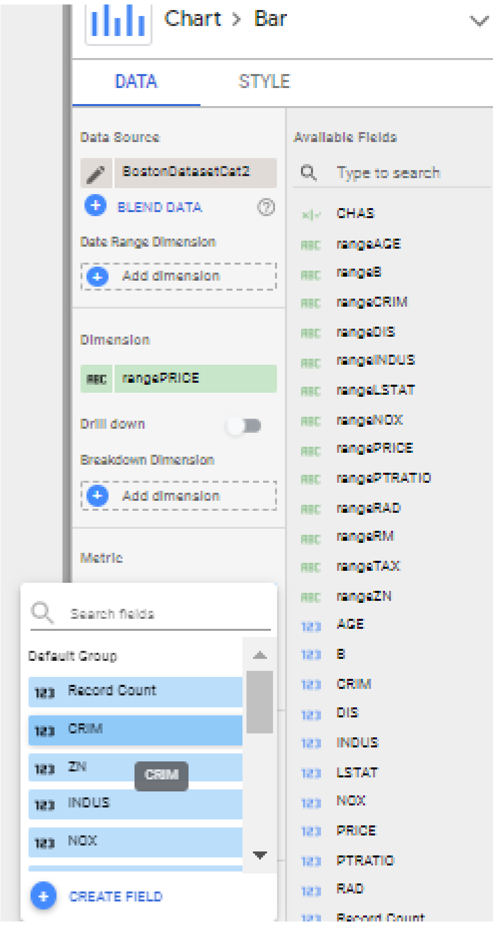


13. Click and drop the box where you would like to place the bar chart on your dashboard.

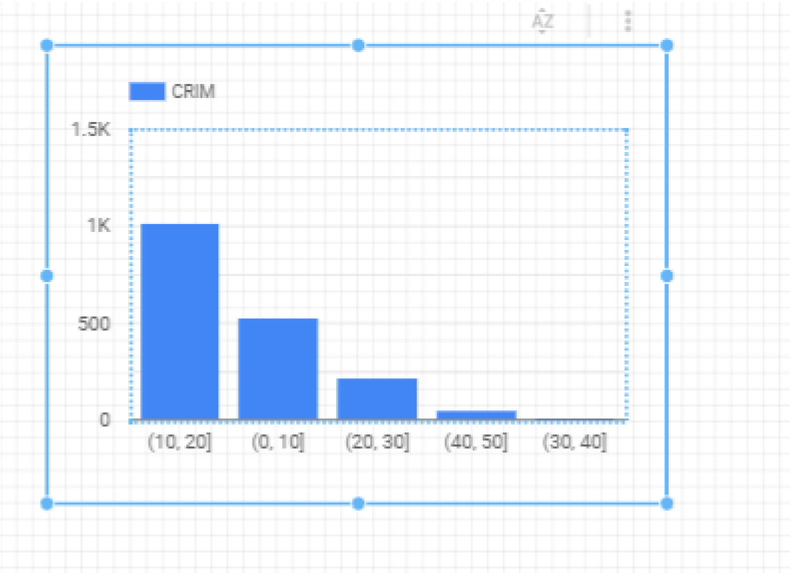
14. Ensure that your bar chart box is selected. Next, click on the green 'CHAS' variable under Dimension in the tool bar on the right hand side and change it to 'rangePrice'.



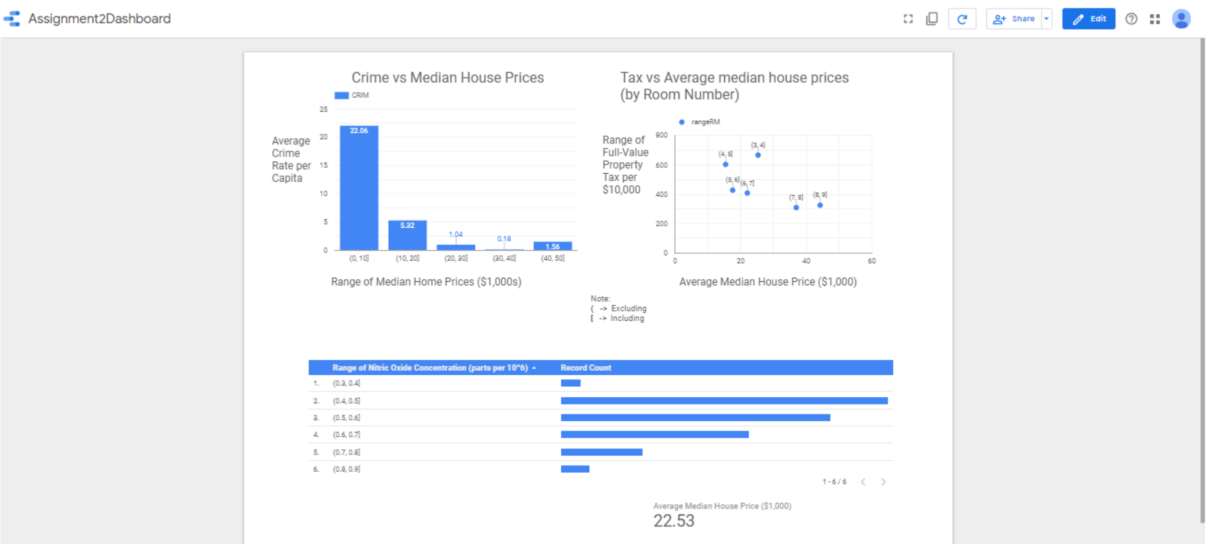
15. Next, make sure the attribute under Metric is 'CRIM'.



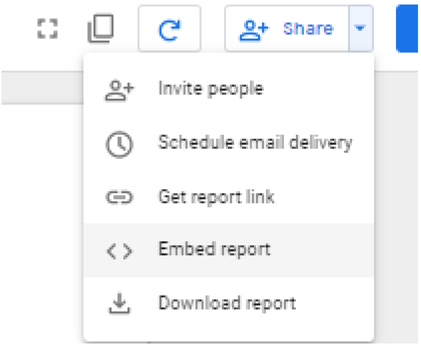
16. Congratulations! You have created your first Data Studio Visualisation! It should look something like this:



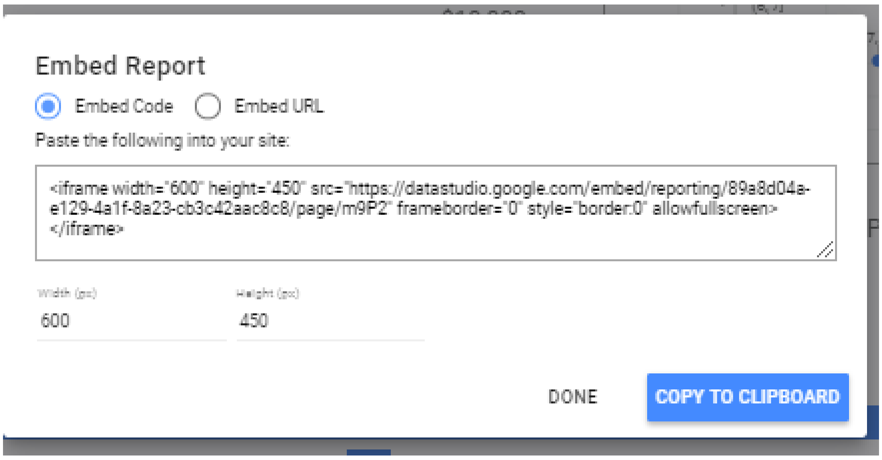
17. You can add text around the graph by clicking on the text box in the top banner. Furthermore, you can also click 'Style' and tick the 'show axis title' for the y and the x axis if you want to automatically generate the axis names. However, in the case of this dataset, the axis names are not the most visually appealing or informative. For this project, they have been manually made using the text boxes.
18. By following these steps, eventually, you can generate a full and informative dashboard. If you click the blue button in the right-hand corner, you can see what the dashboard will look like to users. You can click Edit in the corner again in order to return to the editing stage.



19. Once you are happy with your dashboard, click 'Share' and 'Embed report'



20. This will produce an embedded URL that you paste into your html file for your web-application.



iv. Flask Application

Flask is a python-based web framework that can be used to create a web application. It was used to create the *Boston Housing Price Predictor*. To replicate this application, the user will need to create three python files (application.py, modelling.py and sns.py), one text file (requirements.txt), one json file (credentials.json) and a directory named 'templates' with two html files (main.html and prediction.html). Each file used in this application can be found in the github link (except for credentials.json) in the directory called 'flask'. A description of each file is as follows:

application.py

This file is the main python file for the whole application. This file is used to take the user's inputs for the regression analysis, get the user's phone number for the SNS service, instantiate the model, handle errors with the user's phone number and output the result of the regression. This is also where the application first runs from.

modelling.py

This file handles the regression model for the application. It contains everything that was completed in the Jupyter Notebook (i.e. loading the dataset, preprocessing the data, modelling the data, training the model, etc…). However, this file will also take the user's input values in order to calculate and return the prediction result.

sns.py

This file handles the SNS service for the application. The SNS service will be activated if the developer has created a credentials file and it has a valid AWS access key id and a valid AWS secret access key. If the SNS is successfully activated, the user will receive a text message to the number which they put in the web application with the results of the regression analysis. The application will also display the message 'Message was sent to <phone number>'.

requirements.txt

This file contains all the dependencies required for the application.

credentials.json

This file contains the AWS access key id and secret access key required for the SNS. This file has not been uploaded to the github link for security reasons. However, in order to replicate this file, the developer can create their own. It must be a JSON file.

```
{"access_key_id":"XXXXXXXXXXXXXXXXXX","secret_access_key":"XXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX"}
```

main.html

This html file is the web page for the main page. It handles the font, size and position of every element on the main page. In order to replicate this file, the developer will need to have a heading, body section, each question listed, a text box for the user to put their input, a text box for the user to put their phone number, and two buttons. The first button will submit the user's input for the regression analysis. The second button will show the user the visualisations from the Google Data Studio.

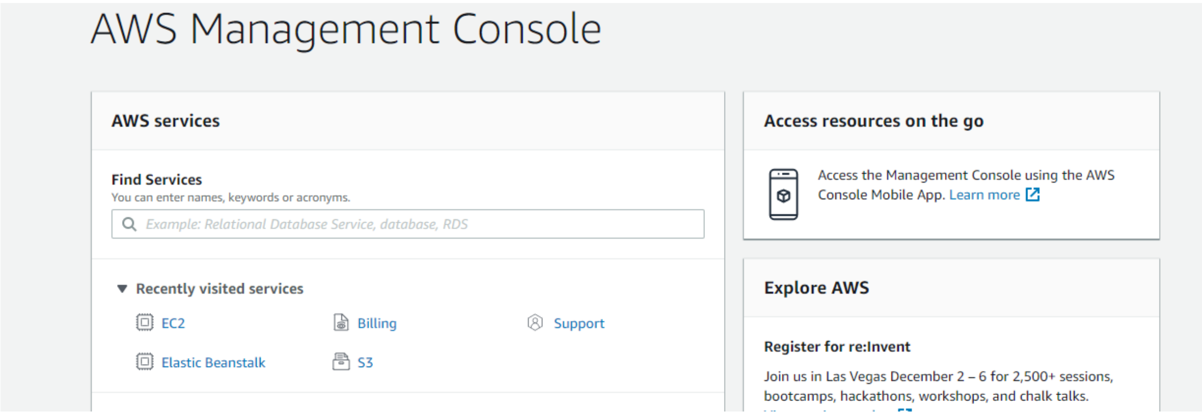
predicition.html

This html file is the webpage for the prediction result. It handles the font, size and position of every element on the prediction page and display's the result.

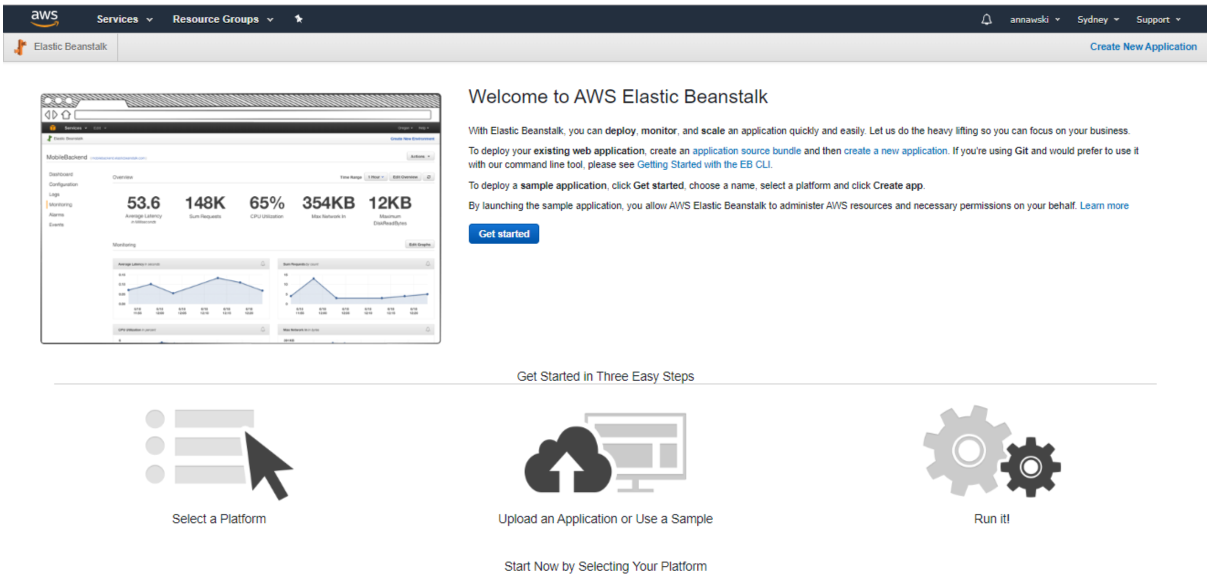
v. AWS

In order to bring the whole project together, the developer must have access to an AWS account. The educate account will not work because it does not give full functionality to the Elastic Beanstalk. The Elastic Beanstalk was chosen for this project because it is a fast and simple way to deploy an application. It will also automatically scale an application up and down depending on the requirements of the application. Follow these steps in order to deploy the application successfully.

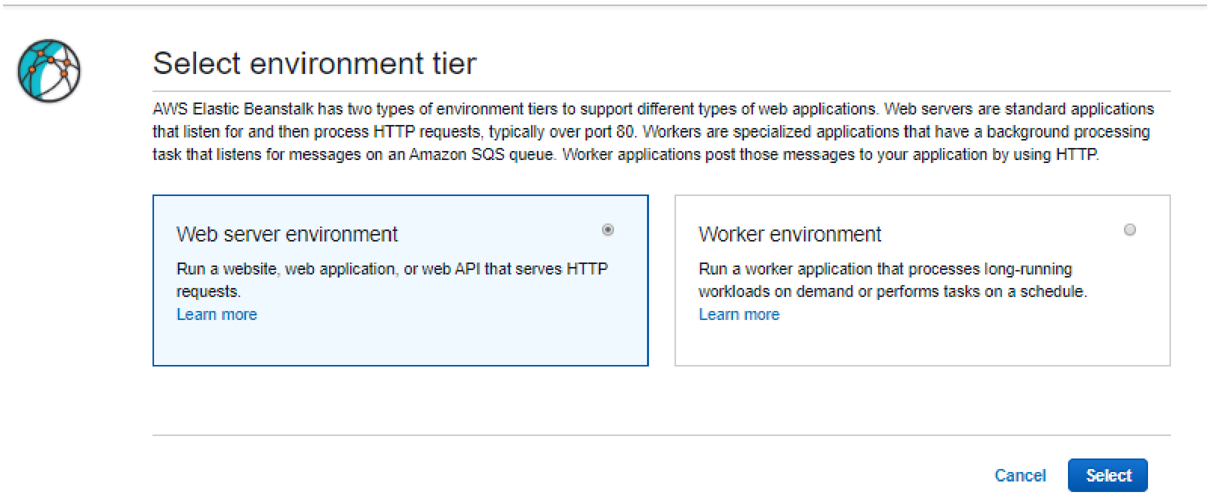
1. Sign into your AWS account: https://aws.amazon.com/
2. Navigate to and click on 'Elastic Beanstalk' from the AWS Management Console page. Search for it under 'Find Services' if you are unable to locate it.



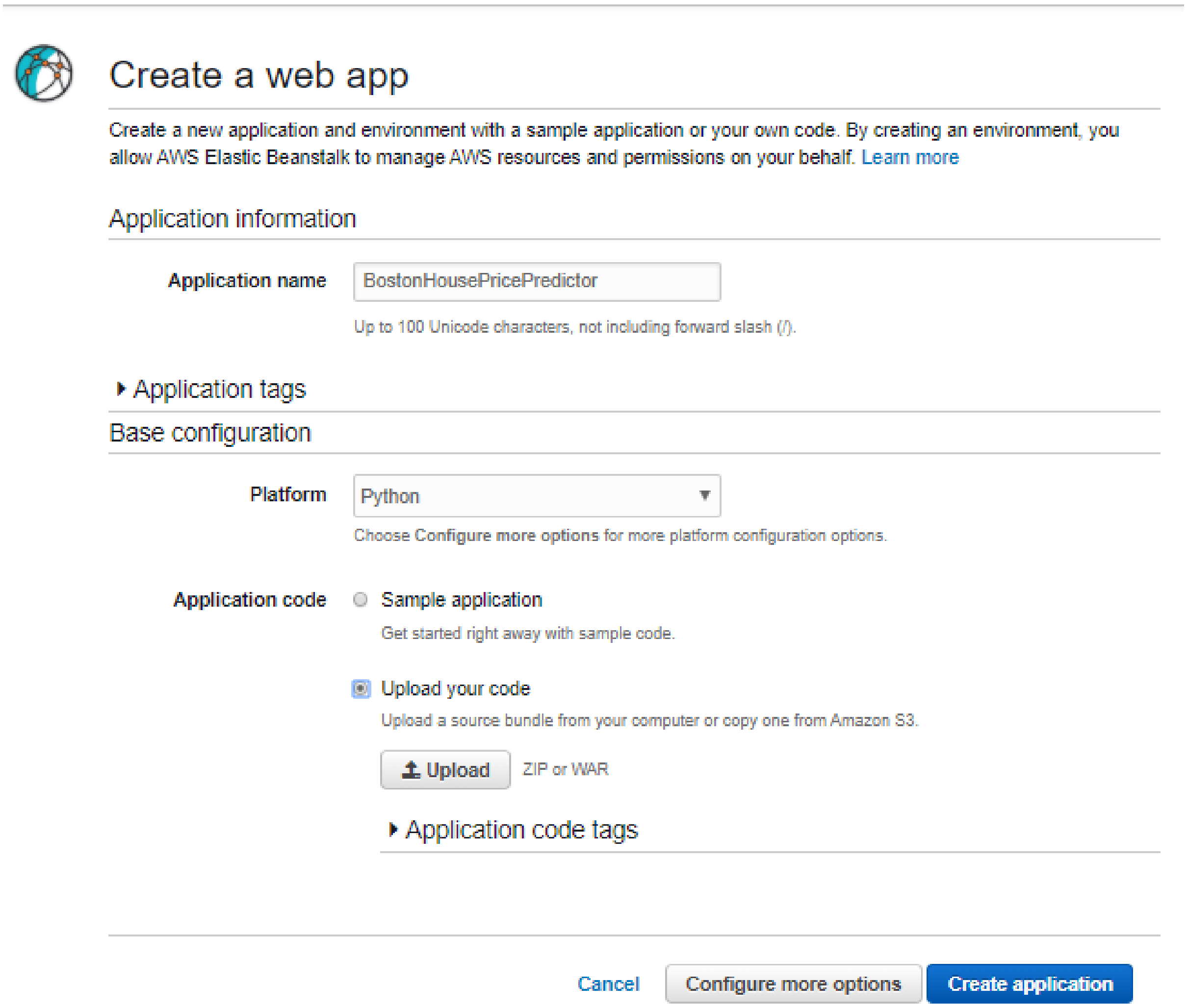
3. Ensure that your location is set to 'Sydney' (top right-hand corner).
4. If you have not created an elastic beanstalk application before, or have deleted all your previous ones, you should see a screen like the one below. Click 'Get started'.



5. Choose Web server environment



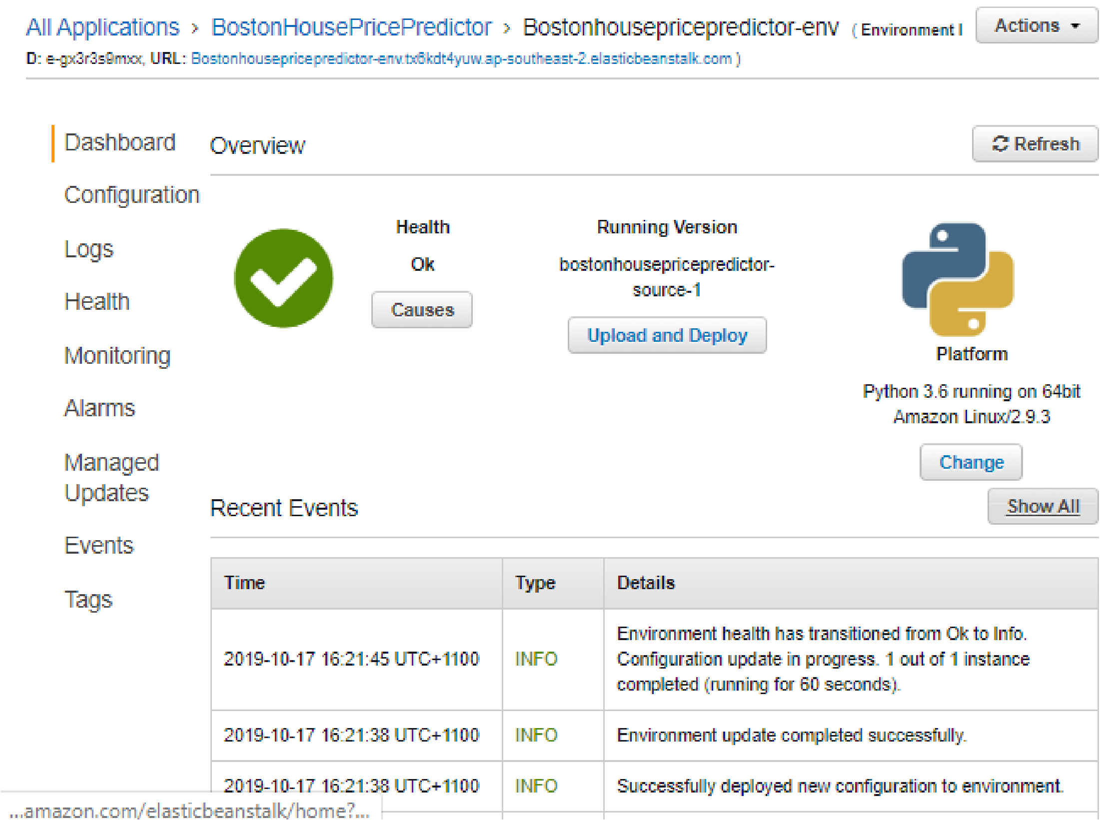
6. Fill out the EB form as follows.



7. Under ‘Application Code’ Click ‘Upload’ and navigate to your flask application’s code. Your flask application must be in a zip file.

8. Click ‘create application’ when this is complete. The creation of the environment should take less than 10 mins.
   NOTE: If you see a WSGIPath error, go to Configuration > Software > Modify. Change the WSGIPath to flask/application.py (or replace flask with the name of the file that your application.py file is in). Then click Apply. You will need to wait again for a few minutes while AWS updates your EB environment. Once this is complete, the Health should have an OK status and you should see this green tick.

9. Next, click on the URL link near the top of the console to see the application:

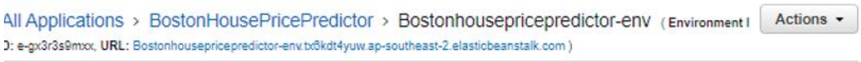


10. Congratulations! Hopefully you have successfully deployed the Boston House Price Predictor Application.

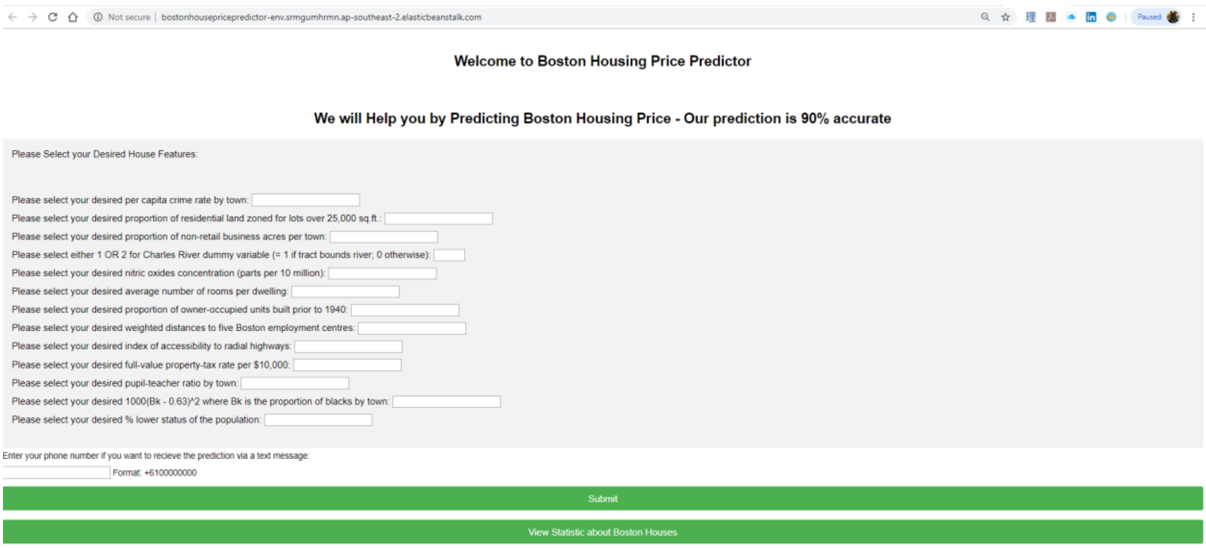
User Manual

This user manual will provide an overview of how to use the Boston Housing Price Predictor Application.

1. Once the application has been successfully deployed to the AWS Elastic Beanstalk, as described in the Implementation section of this report, the application can be accessed from the URL generated at the top of the beanstalk's dashboard.



2. The user will be taken to the main page when they click this link. There is a list of questions for the user to read through. The user's answers to these questions will determine the output of the regression analysis.



3. The user must put in a positive numerical value. The user may also put their mobile number in the relevant text box if they would like to receive a text message. The following example is what the user will see if they do not put in their phone number.

Please select your desired per capita crime rate by town: 8

Please select your desired proportion of residential land zoned for lots over 25,000 sq.ft.: 8

Please select your desired proportion of non-retail business acres per town: 8

Please select either 1 OR 2 for Charles River dummy variable (= 1 if tract bounds river; 0 otherwise): 1

Please select your desired nitric oxides concentration (parts per 10 million): 8

Please select your desired average number of rooms per dwelling: 8

Please select your desired proportion of owner-occupied units built prior to 1940: 8

Please select your desired weighted distances to five Boston employment centres: 8

Please select your desired index of accessibility to radial highways: 8

Please select your desired full-value property-tax rate per $10,000: 8

Please select your desired pupil-teacher ratio by town: 8

Please select your desired 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town: 8

Please select your desired % lower status of the population: 1

Enter your phone number if you want to recieve the prediction via a text message:

Format: +6100000000

Welcome to Boston Housing Price Predictor

Predicted Median value of owner-occupied homes in $1000's is: 39.93818862

4. If the user does put in their phone number (in the format described below), the application will confirm that a message was sent to the user's phone number.

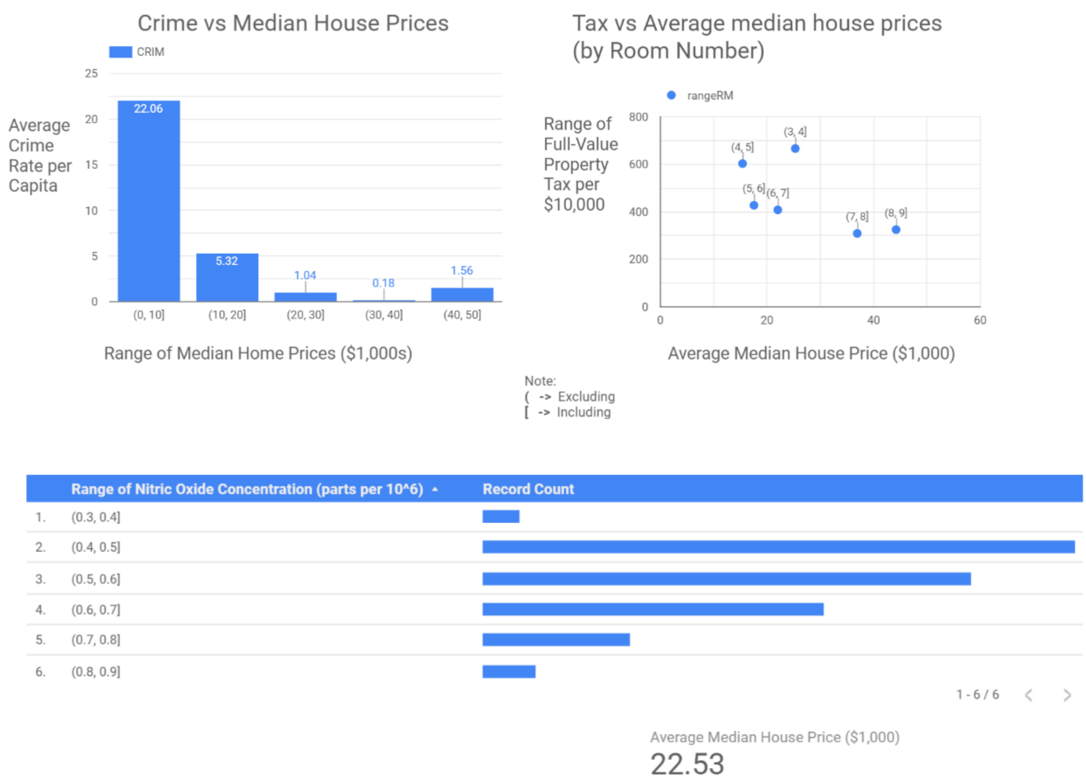
Welcome to Boston Housing Price Predictor

Predicted Median value of owner-occupied homes in $1000's is: 39.57656603

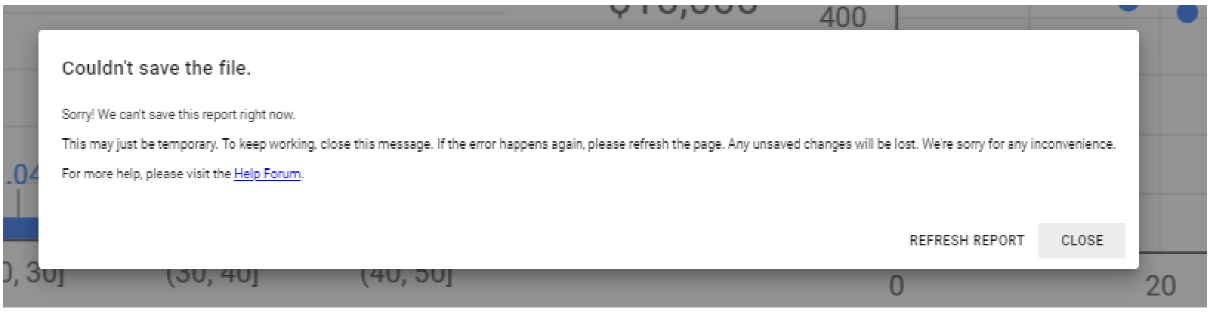
message was sent to +61409896140

+61409896140 Format: +6100000000

5. If the user clicks the last button, ‘View Statistic About Boston Houses’, they will be taken to an interactive visualisation of the Boston Housing Dataset. This visualisation is not dependent upon the user’s inputs. It is designed to act as a summary for user’s who are not sure what to expect with the dataset. It is designed to give them an overall understanding of the housing prices in Boston.



6. Lastly, when visiting this interactive site, the user may come across this message. This message is a known bug with Google Data Studio. As the user is not saving the file, this message will not affect the user visualising and interacting with the visualisation. The user can click ‘CLOSE’ to look at the visualisation.



References

[1]"Getting Started Using Elastic Beanstalk - AWS Elastic Beanstalk", *Docs.aws.amazon.com*, 2019. [Online]. Available: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/GettingStarted.html. [Accessed: 19- Oct- 2019].

[2]"About sharing - Data Studio Help", *Support.google.com*, 2019. [Online]. Available: https://support.google.com/datastudio/answer/6287179. [Accessed: 19- Oct- 2019].

[3]"BigQuery documentation|BigQuery|Google Cloud", *Google Cloud*, 2019. [Online]. Available: https://cloud.google.com/bigquery/docs/. [Accessed: 19- Oct- 2019].

[4]M. Lacey, "Multiple Linear Regression", *Stat.yale.edu*, 2019. [Online]. Available: http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm. [Accessed: 19- Oct- 2019].

[5]"sklearn.datasets.load_boston — scikit-learn 0.21.3 documentation", *Scikit-learn.org*, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html. [Accessed: 19- Oct- 2019].

[6]"What is Amazon Simple Notification Service? - Amazon Simple Notification Service", Docs.aws.amazon.com, 2019. [Online]. Available: https://docs.aws.amazon.com/sns/latest/dg/welcome.html. [Accessed: 19- Oct- 2019].

[7]"Welcome to Flask — Flask Documentation (1.1.x)", Flask.palletsprojects.com, 2019. [Online]. Available: http://flask.palletsprojects.com/en/1.1.x/. [Accessed: 19- Oct- 2019].

[8]"Understanding property prices", Consumer.vic.gov.au, 2019. [Online]. Available: https://www.consumer.vic.gov.au/housing/buying-and-selling-property/buying-property/understanding-property-prices. [Accessed: 19- Oct- 2019].

[9]"Real Estate | Properties for Sale, Rent and Share | Domain", Domain, 2019. [Online]. Available: https://www.domain.com.au/. [Accessed: 19- Oct- 2019].