Udacity Machine Learning Nanodegree 2021

Capstone Project Report

Predicting hospital bed demand in Australia with Machine Learning

Ryan Ashton

Introduction

This project report aims to address all 5 stages outlined in the original proposal document. Followed by a section on results and areas of improvement.

Stage 1: Collecting the data

This is finding as many data sources as possible related to Australian hospitals and demographics and joining these datasets together.

This stage was conducted in the Jupyter Notebook named, "Data_Preprocessing.ipynb", which reads in all the different data sources, cleans, and joins the data into a final output to be used for the EDA Jupyter Notebook.

The data can be found in the folder named, "Data" which have come from different sources.

The following datasets come from the Australia Bureau of Statistics:

- education_level_age_SA2
- employment_by_age_SA2
- indigenous_population_SA2
- mortgage_repayments
- Number_of_Dwellings
- personal_income_Australia_SA2
- Population_age_sex
- renting_SA2
- SA2_Area_KM

The mapping file "SA2_Postcode_Map" was requested from the Australian Government – fortunately, they were happy to provide this data as this creates the opportunity to join demographic data with hospital bed data.

The file "Public_Hospitals_with_postcode" is a file that can be publicly obtained from the Australian Institute of Health and Welfare, however, the additional postcode column was manually mapped by myself so that the joins can be conducted – this manual mapping time investment gives the unique power of the dataset (the ability to link hospital data with SA2 Census data).

Each dataset must have at least a postcode or an SA2 (Statistical Areas Level 2). An SA2 is a geographical area that represents a community that interacts together socially and economically. The SA2 level data generally has the most features publicly available at the lowest granular level of geographical area.

This means that we can take the hospitals at postcode level with their respective available beds and aggregate the number of beds to the corresponding SA2 level.

A visual representation is below:

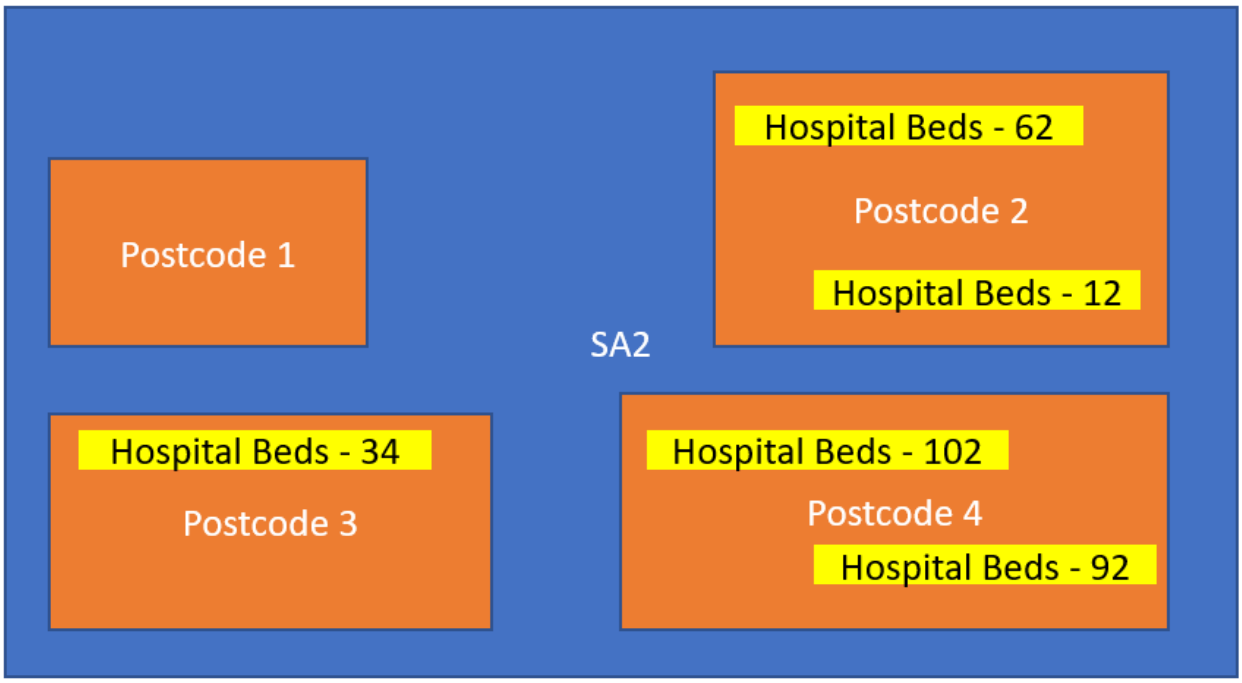


Figure 1: An SA2 showing a total of 302 available hospital beds (62+12+34+102+92).

What I did find, however, is that when the Australian Government provided the SA2 to Postcode mapping, a Postcode can be within more than one SA2 (as illustrated below):

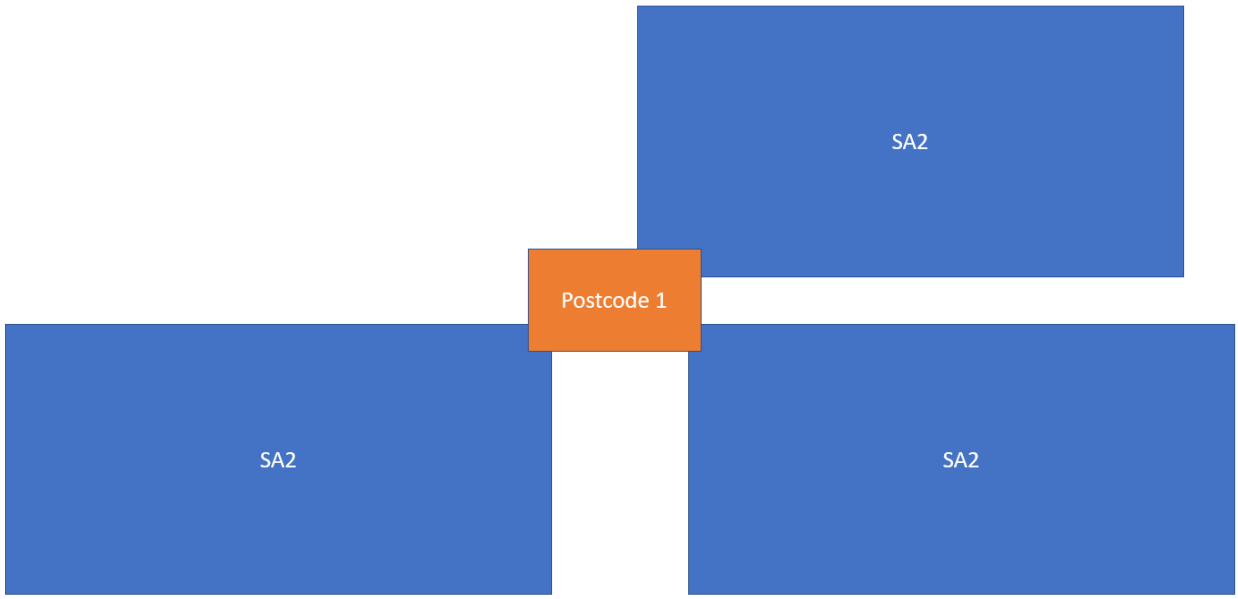


Figure 2: A potential anomaly which shows that 1 postcode can belong to multiple SA2s.

Whilst this is not ideal, for the purpose of this capstone the hospital beds from 1 postcode will be aggregated across different SA2 regions.

Stage 2: Exploratory Data Analysis

As there will be a lot of useful data collected here, it is worth exploring the data which could also be used for the web application.

This stage was conducted in the Jupyter Notebook named, "EDA.ipynb", which reads in the final data output from the Data_Preprocessing.ipynb Jupyter Notebook located in the folder "Output_Data".

As a first step, the basic health of the dataset was checked to see if it could be worked with. This included:

- The shape of the dataframe
- If there are any nulls
- The number of unique values in each feature
- The datatypes
- The statistical overview with Pandas' .describe()

The target we are trying to predict is the number of available hospital beds by a SA2 region. The current distribution of hospital beds across regions appears to be positively skewed:

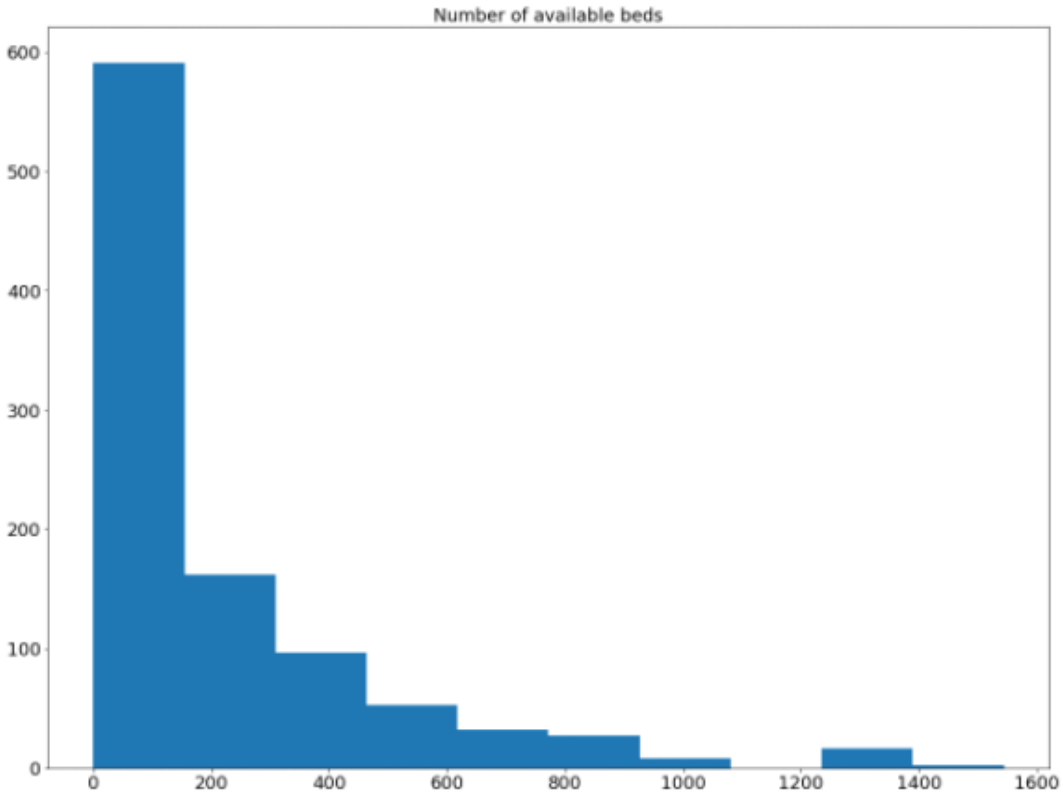


Figure 3: A histogram of the hospital bed numbers across SA2s.

The features that were obtained from the pre-processed dataset were the following:

- Population of the area
- Median income of the area
- Number of Dwellings within the area
- How many people were working/earning an income in the area
- Number of people who were tertiary educated in the area
- How many people were paying a mortgage in the area
- How many people were renting their place in the area
- How many indigenous people lived in the area

Because the "totals" were used for most of these features, not many deep dives were required for the purpose of this Capstone assignment. This also means that at this stage, only the calculator will be available on the Web App.

The initial development of this dataset was designed for the purpose to be trained on. Therefore, the correlations with the available hospital beds were the next important step before developing the Machine Learning model.

Stage 3: Pre-Processing / Feature Engineering and Selection

This stage will be for preparing the final inputs (training and test data) prior to training the Machine Learning model

This stage was conducted in the Jupyter Notebook named, "EDA.ipynb", which reads in the final data output (new_df.xlsx) from the Data_Preprocessing.ipynb Jupyter Notebook located in the folder "Output_Data".

A heat map for the correlations was used with the Number of Available (hospital) beds being the focus as it was the target to predict. Here is the heat map prior to removing the anomalies:

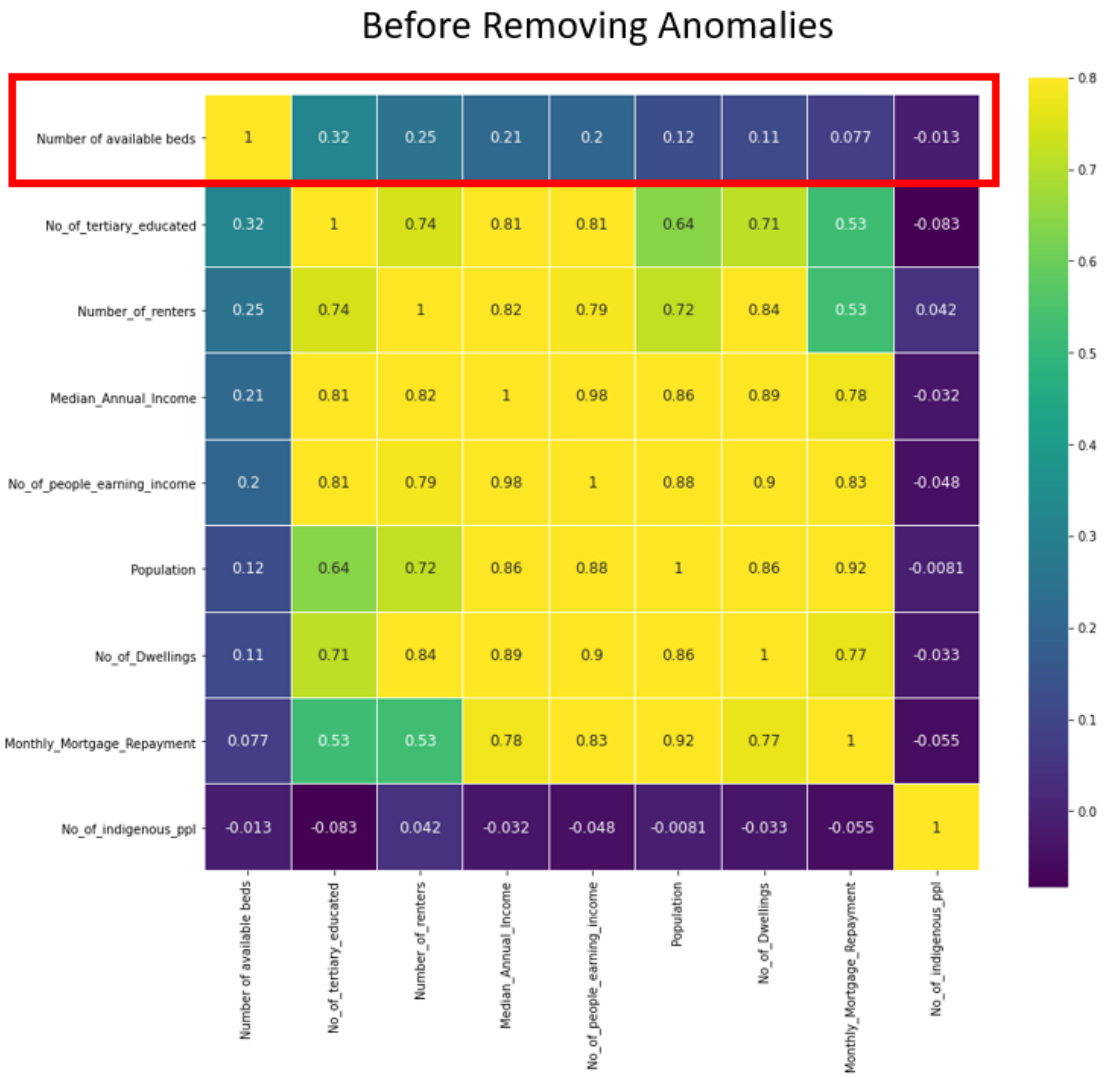


Figure 4: A heat map illustrating how the features correlate with the number of hospital beds by SA2.

It was surprising to see these correlations as I had anticipated that population would be highly correlated with the number of hospital beds required. I found that Central Business Districts (CBD) were affecting this correlation, as the high-density populated areas did not necessarily have the largest hospitals in the area. Similarly, the very low-density populated areas had a similar effect on the dataset.

I tried introducing another feature to the dataset to see if geographical size was influencing the correlations. Once brought in, I decided it had no impact. However, the population correlation still needed to be addressed.

For this reason, I removed data that had the following characteristics shown in the visual below:

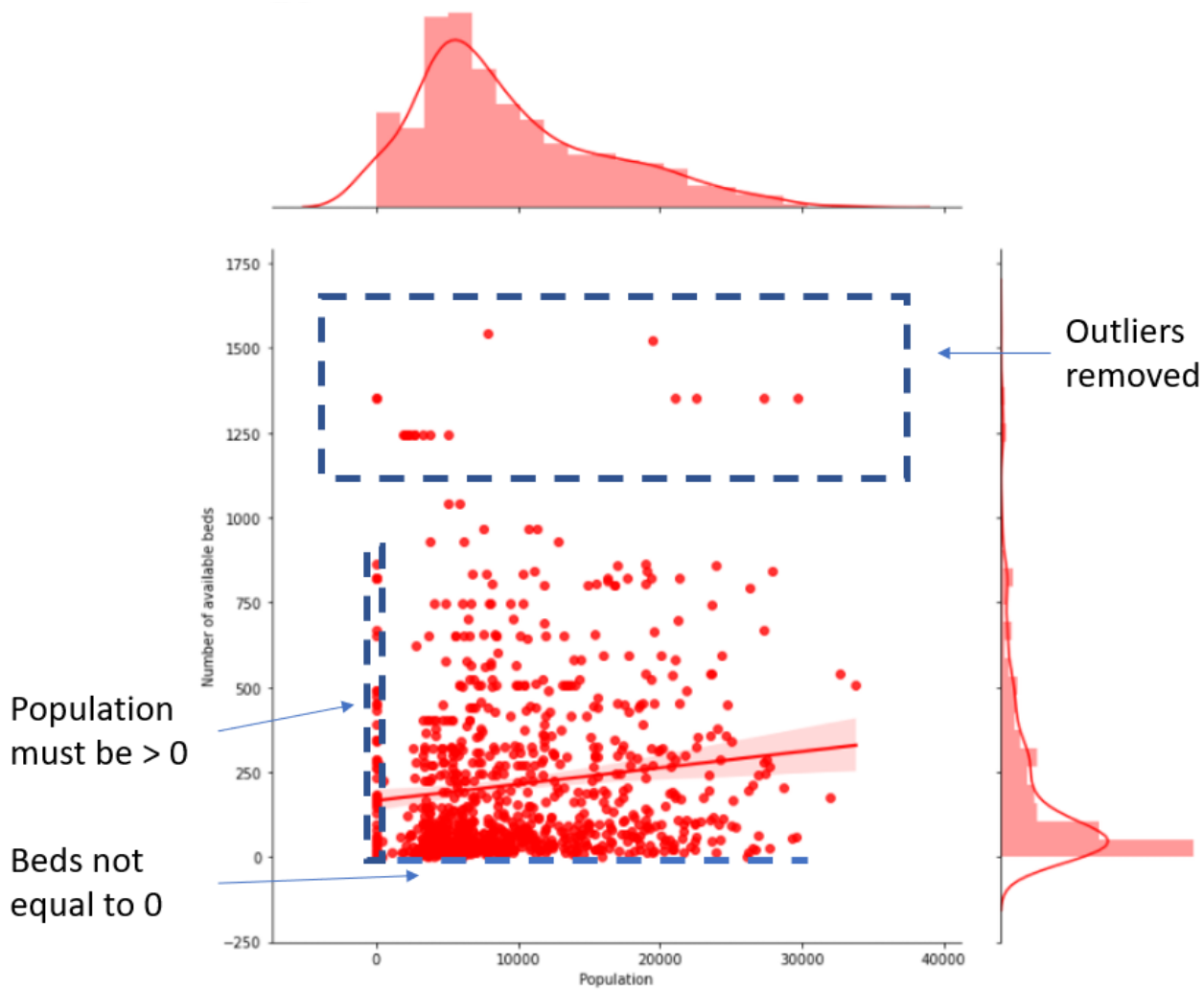


Figure 5: A joint plot with highlights of where the dataset can be optimized for modelling.

This produced a stronger correlation overall with the heat map showing the following:

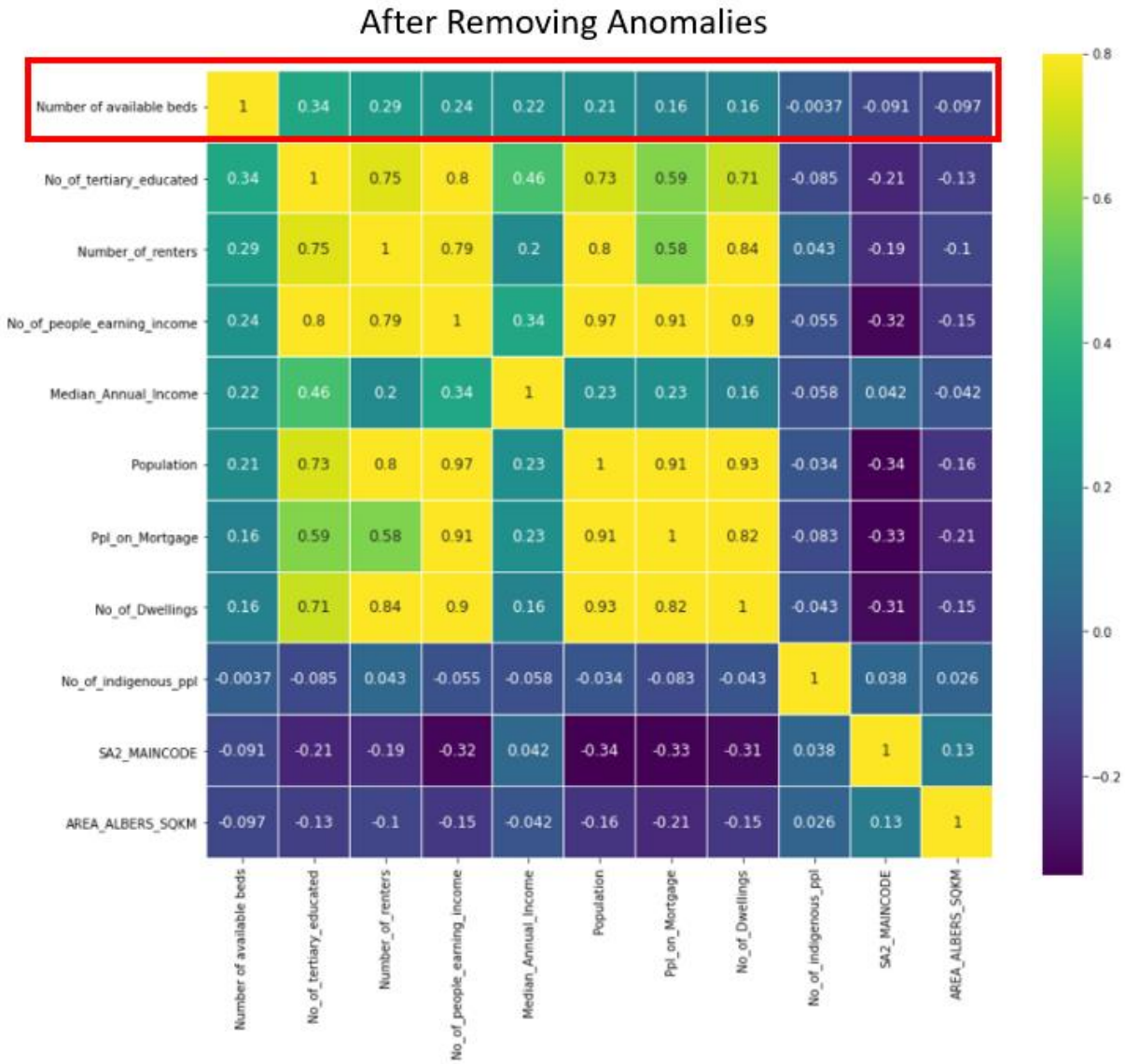


Figure 6: The revised heat map illustrating how the features correlate with the number of hospital beds by SA2 after the outliers were removed.

After many iterations with different features and considerations of end-user experience with the Machine Learning App, the following features were selected to predict hospital beds (determined in ML.ipynb):

- Population
- Number of Dwellings
- Median Annual Income
- Number of people who are tertiary educated (highest correlation)

Stage 4: Selecting a Machine Learning model and Hyperparameter Tuning Many regression models will be tested with the data to find the most suitable algorithm (without overfitting).

This stage was conducted in the Jupyter Notebook named, "ML.ipynb".

Because the prediction is to determine the number of hospital beds required based on numeric demographic assumptions, the algorithms to be considered will be for regression.

Algorithms and Techniques

The following regression algorithms tested were:

- Random Forest Regressor
- Multiple Linear Regression
- Decision Tree Regressor
- Support Vector Regressor

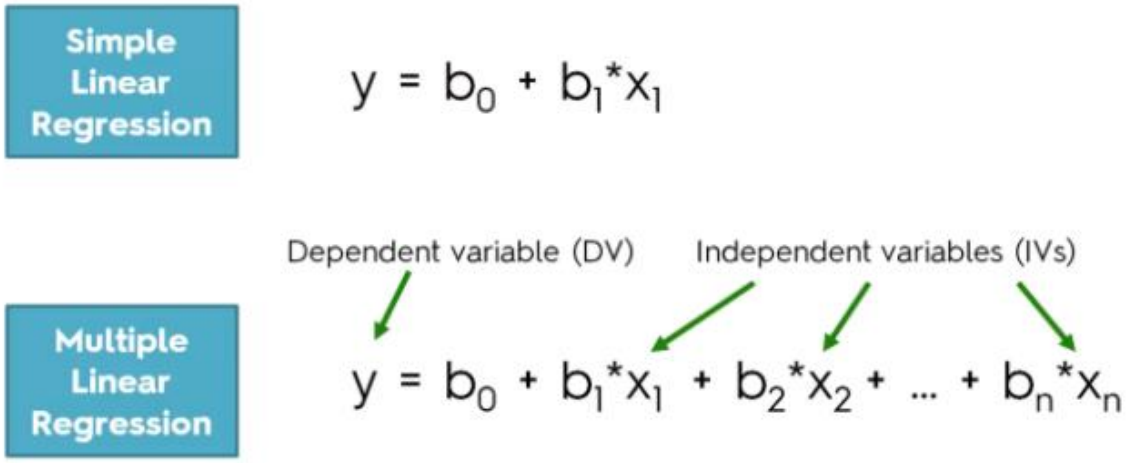
Random Forest Regressor

The Random Forest Regressor is a powerful Machine Learning algorithm because it utilises ensemble learning, meaning, it creates multiple models and then combines them to improve the result. This is performed with a tree structure with each branch having its own test and prediction. All the trees' predictions are averaged to then give a Random Forest Prediction.

Multiple Linear Regression

Because there are multiple features utilised for the prediction of Hospital beds there needs to be an equation that extends beyond normal linear regression (which considers only one feature or variable). Multiple Linear Regression therefore mathematically extends the independent variables by addition.

A visual example is provided by the Medium post (https://medium.com/@manjabogicevic/multiple-linear-regression-using-python-b99754591ac0):



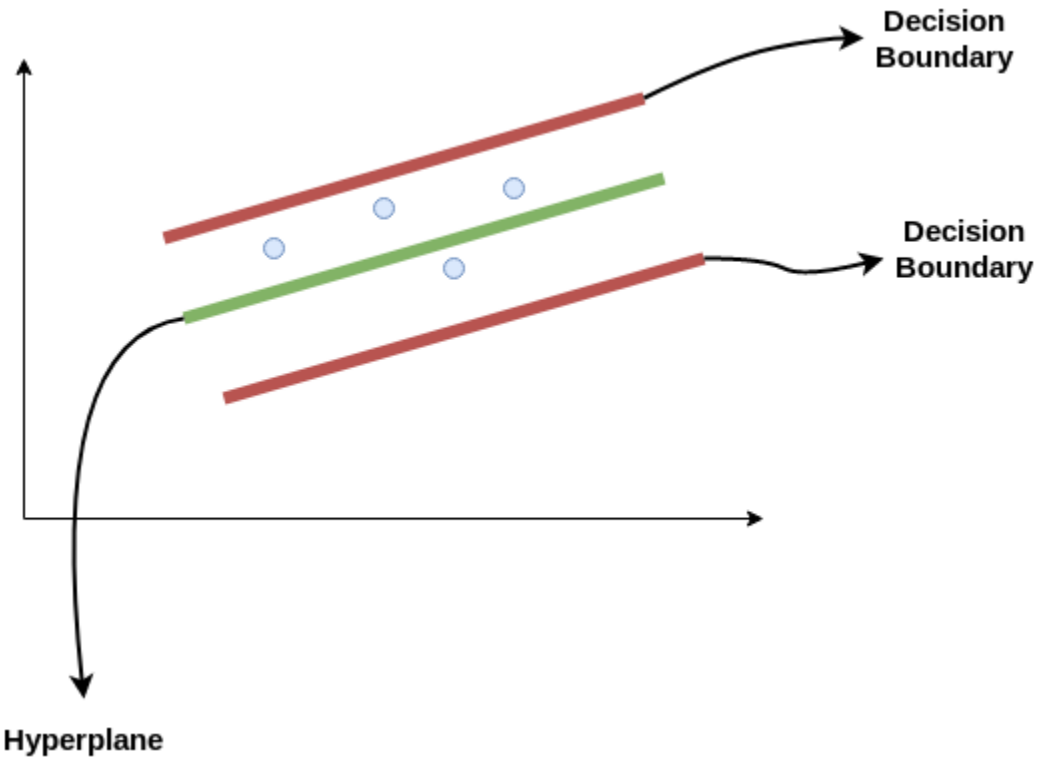
Decision Tree Regressor

A Decision Tree Regressor behaves as multiple "if statements" based on the data provided to it. It will consider the most optimal choice within each node of the tree. For this reason, it is often prone to over-fitting as it does not want to look at sampling less optimal paths.

Support Vector Regressor

The Support Vector Machine algorithm is often used more for classification problems, but it can still be used for regression problems. The algorithm acts like Multiple Linear Regression, but it can consider data points that deviate away from the line of best fit (called a Hyperplane). This allowance for deviation is created by boundaries, called "Decision Boundary" on each side of the Hyperplane.

A visual example is provided by the following website (https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/):



Testing Performance

The results of testing each algorithm were:

Model	Score
DecisionTreeRegressor	100.00
RandomForestRegressor	87.69
LinearRegression	13.94
SVR	-18.11

The Random Forest Regressor was chosen as it showed the highest score without indicating that there was overfitting. As mentioned previously, the Decision Tree Regressor was prone to over-fitting which was seen in my tests.

Benchmark Research

Three research papers were identified to help determine a benchmark for performance. The titles of these papers were:

- Predictive Models for Hospital Bed Management using Data Mining Techniques (https://core.ac.uk/download/pdf/55630976.pdf)

- COVID-19: Short-term forecast of ICU beds in times of crisis (https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0245272)

- Length of Hospital Stay Prediction at the Admission Stage for Cardiology Patients Using Artificial Neural Network (https://www.hindawi.com/journals/jhe/2016/7035463/)

Table of Benchmark Comparisons:

Research Title	Problem to Solve	Algorithm/Models Used	Accuracy
Predictive Models for Hospital Bed Management using Data Mining Techniques	Predict the number of patient discharge (leaving hospital) - which can provide an input to predict number of hospital bed requirements	Decision Trees, Naïve Bayes, Support Vector Machine	≈82.69% to ≈94.23%
COVID-19: Short-term forecast of ICU beds in times of crisis	Forecast demand for ICU capacity due to Covid-19	Autoregressive, Machine Learning (undefined) and Epidemiological models	Forecasting errors of 4% and 9%
Length of Hospital Stay Prediction at the Admission Stage for Cardiology Patients Using Artificial Neural Network	Predict how long a cardiology patient will stay in hospital	Artificial Neural Net / Linear Regression	Pre-discharge stage - 88.07% to 89.95% Pre-admission stage - 88.31% to 91.53% Heart Failure / Myocardial Infarction patient - 63.69% to 67.47%
Udacity Capstone	**Predict number of hospital beds for new populated area in Australia**	**Random Forest Regressor**	**87.69%**

Stage 5: Deploying the model / Web Application

The model will be deployed as a Dash and Plotly App with Heroku. This will allow the user to provide inputs so they can simulate how many hospital beds are required in a new area.

The web application is live at this url: https://udacity-hospital-bed-predict.herokuapp.com/

Results and Areas of Improvement

What worked well?

This project was developed completely from the start, with my own ideas. There were many things I was able to accomplish such as creating the dataset with many joins, performing EDA as well as developing and tuning the machine learning model. The experience of working end to end on a project like this was very rewarding as I was able to learn a lot during the process.

What could be improved?

Because this project was the first of its kind, I encountered a lot of errors and doubts along the way - which wasted a lot of time. Although the Machine Learning model was able to score highly with the Random Forest Regressor, the correlations between the features and target was not as high as I expected. After this submission has been considered completed by Udacity, I intend on revisiting this web app again in the future and look to see if I can find other features to help forecast hospital bed numbers for new developing suburbs.

The Result and Benchmark Comparison

Although this Machine Learning model scored within the same range as other research papers outlined in "*Table of Benchmark comparisons*" I do not believe this web app is accurate enough to send to the Australian Government for their main forecasting tool. This is due to the lack of coordination with the Australian Government itself and comparing this model with their current models. However, I do believe it is a particularly good start.

The methodology of building this Web App is much stronger than the result itself. Overall, this model can continue to be improved and perhaps there might be an opportunity to expand the App's scope into something the Australian Government can use one day.