Melbourne Property Predictor

By Catherine Sloan, Tom Peddlesden, Danielle Cahill, Anne Wieggers and Joe Quinn

THE PROBLEM

TONY IS STRESSED

OUR SOLUTION

With many current tools, they only show the asking price of the home or the current estimated value.

We wanted to create a tool that not only shows the current price, but also a prediction of the property's future value using machine learning.

It also doesn't require users to input their personal information or pay for a valuation.

Best of all…..IT'S FREE!!

CONSIDERATIONS

- ➔ Number of Bedrooms
- ➔ Number of Bathrooms
- ➔ Car Spaces
- ➔ Property Size
- ➔ Distance from CBD
- ➔ Historical Price Data
- ➔ Crime statistics
- ➔ Month
- ➔ Year

HOW IT WORKS

1. User finds a property listing they like on Domain
2. Copy the URL and paste it into our predictor
3. Predictor scrapes the details from the URL
4. Details are run through machine learning model
5. Predictor gives an estimated price

WHAT WE USED

- Python
- Javascript
- Flask
- Beautiful Soup and Requests
- Machine Learning Model xgboost
- Sklearn
- HTML
- CSS
- Bootstrap
- Heroku

Machine Learning

Data Sources

- House price data set (Kaggle)
- Victorian crime data (Data.vic.gov)
- Average yearly property price increase (Data.vic.gov)

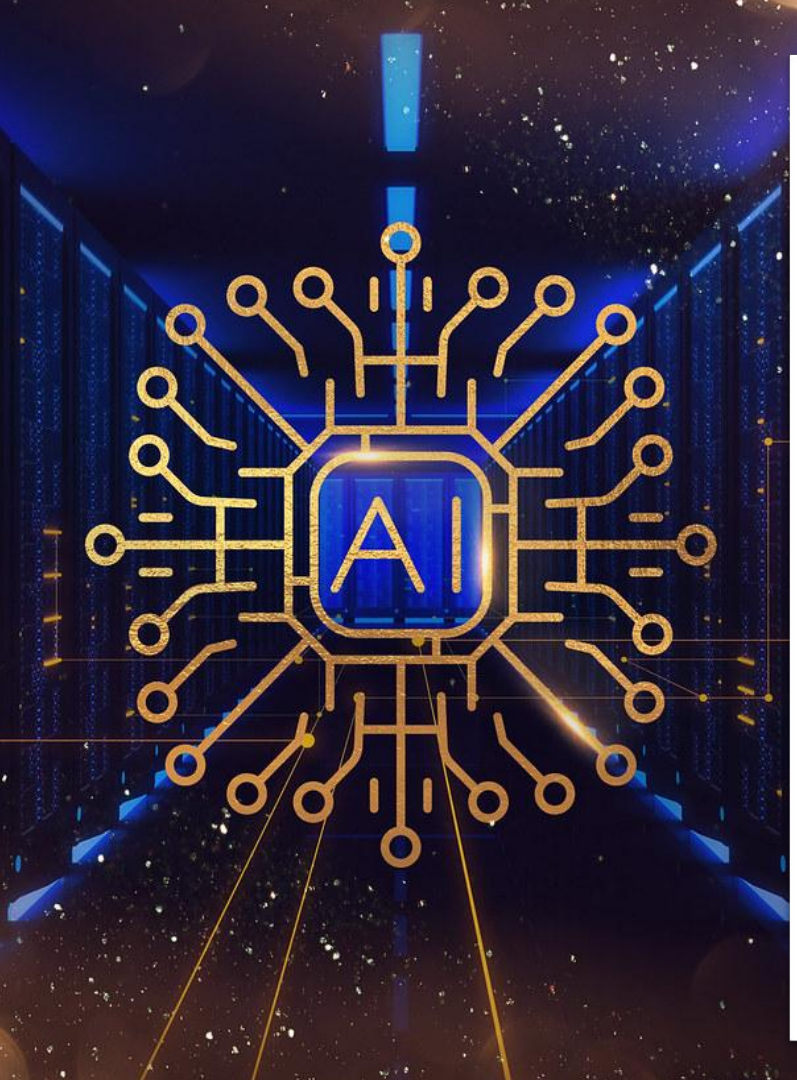
Data Preprocessing

- Merging Housing data and Crime data
- Removing unnecessary columns and NaN values. Included variables must be:
  - Available on Domain
  - Reasonably assumed to influence property prices
- Changing data types and datetime variables
- Binary Encoding the Type variable
- Scaling the X variables to normalise before modelling

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Bathroom	Car	Landsize	BuildingArea	YearBuilt
0	Abbotsford	68 Studley St	2	h	NaN	SS	Jellis	3/09/2016	2.5	3067.0	...	1.0	1.0	126.0	NaN	NaN
1	Abbotsford	85 Turner St	2	h	1480000.0	S	Biggin	3/12/2016	2.5	3067.0	...	1.0	1.0	202.0	NaN	NaN
2	Abbotsford	25 Bloomburg St	2	h	1035000.0	S	Biggin	4/02/2016	2.5	3067.0	...	1.0	0.0	156.0	79.0	1900.0
3	Abbotsford	18/659 Victoria St	3	u	NaN	VB	Rounds	4/02/2016	2.5	3067.0	...	2.0	1.0	0.0	NaN	NaN
4	Abbotsford	5 Charles St	3	h	1465000.0	SP	Biggin	4/03/2017	2.5	3067.0	...	2.0	0.0	134.0	150.0	1900.0

	postcode	suburb	lat	lon	Local Government Area	Region	Year	A20	A50	A70	...	F20	F30	F90	Total	A	B	C	D	E	F
0	3000	melbourne	-37.814563	144.970267	Melbourne	Northern Metropolitan	2011	1032	116	99	...	13	36	3	14175	1414	7331	404	3764	1210	52
1	3002	east melbourne	-37.816640	144.987811	Melbourne	Northern Metropolitan	2011	53	12	4	...	0	9	0	753	76	476	32	149	11	9
2	3003	west melbourne	-37.806255	144.941123	Melbourne	Northern Metropolitan	2011	54	9	3	...	2	1	2	633	80	403	32	107	6	5
3	3006	southbank	-37.823258	144.965926	Melbourne	Southern Metropolitan	2011	237	21	14	...	0	14	2	2059	310	1103	60	545	25	16
4	3008	docklands	-37.814719	144.948039	Melbourne	Southern Metropolitan	2011	113	7	8	...	4	6	3	1244	149	641	35	389	17	13

	Rooms	Price	Distance	Bathroom	Car	Landsize	Year	Month	Crime	Type_h	Type_t	Type_u
0	0.090909	1480000.0	0.05	0.111111	0.055556	0.000466	0.0	0.181818	0.068001	1	0	0
1	0.090909	1035000.0	0.05	0.111111	0.000000	0.000360	0.0	0.272727	0.068001	1	0	0
2	0.272727	1600000.0	0.05	0.111111	0.111111	0.000277	0.0	0.272727	0.068001	1	0	0
3	0.090909	941000.0	0.05	0.111111	0.000000	0.000418	0.0	0.545455	0.068001	1	0	0
4	0.181818	1876000.0	0.05	0.222222	0.000000	0.000566	0.0	0.545455	0.068001	1	0	0

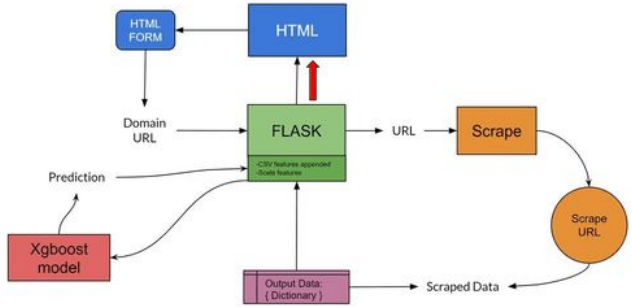


MODELLING

- XGBRegressor model from the XGBoost library
  - A gradient boosting decision tree model
- **Model 1**: trained on Bedrooms, Bathrooms, Car Spaces, Property Type (house, unit, townhouse), Land Size, Year, Month, Crime Rate.
- **Model 2:** trained on Bedrooms, Bathrooms, Car Spaces, Property Type (house, unit, townhouse), Year, Month, Crime Rate for when Land Size is not available on Domain.com
- Hyperparameter Optimisation improved the models by 1.54% and 1.60% respectively.

Backend

Flow Chart



HTML FORM

HTML

Domain URL

FLASK

-CSV features appended
-Scale dictionary features

URL

Scrape

Prediction

Scrape URL

Xgboost model x 2

Property Features Scaled

Output Data: { Dictionary }

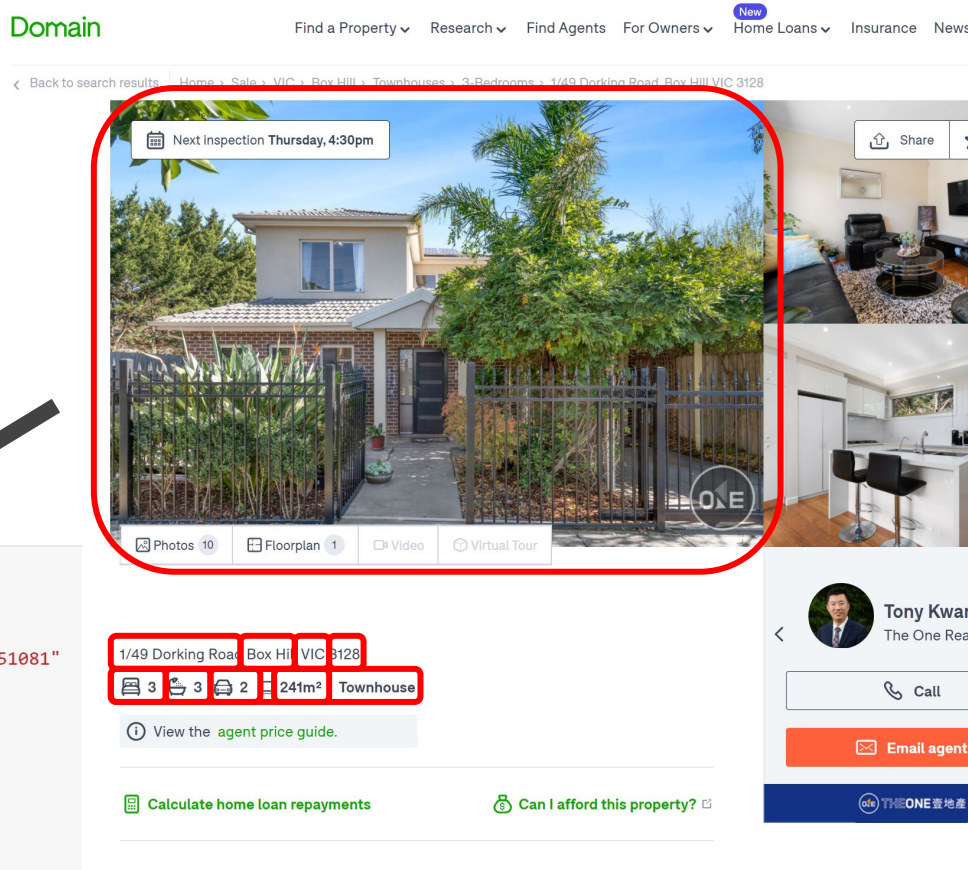
Scraped Data

Steps undertaken

1. Scrape Domain URL
2. Append CSV data for suburbs
3. Binary format and scale data
4. Predict using model
5. Render HTML with results with scraped Data

Web Scraping

- Scraped www.domain.com.au listings with the combination of Selenium and BeautifulSoup
  - Selenium to automate web browser interaction
  - BeautifulSoup to load the page source




```
def initialise_browser():
    executable_path = {'executable_path': ChromeDriverManager().install()}
    return Browser("chrome", **executable_path, headless=False)
browser = initialise_browser()
url = "https://www.domain.com.au/970-drummond-street-carlton-north-vic-3054-2016951081"
# retrieve page with the requests module and visit url
response = requests.get(url)
browser.visit(url)
# getting the html code of browser
html = browser.html
print(html)
# parse object with BeautifulSoup as bs
soup = bs(html, 'html.parser')
summary = soup.find("div", class_="css-fnm9v")
```

Matching to more features

```
# -------------------- CRIME AND DISTANCE FOR SUBURB AVERAGE
# read in csv file and find distance and crime based on subu

def distance_crime(suburb_lower):

    suburb_c_t = pd.read_csv("data/suburb_crime_distance_a
    #loop through rows
    for index, row in suburb_c_t.iterrows():
        # find row that matches scrapes suburb
        if (suburb_lower == row["Suburb"]):
            # store row values
            distance = row["Distance"]
            crime  = row["Crime"]
            avg_increase = row["Average_increase"]

    return(distance, crime, avg_increase)
```

ta > suburb_crime_dist.csv

```
1  Suburb,Distance,Crime
2  abbotsford,2.5,1053.0
3  airport west,13.5,737
4  albert park,3.3,572.0
5  alphington,6.4,209.0
6  altona,13.8,613.0
```

Convert String Values and Scale

```
# and convert to numerical value
convert_type = features["ptype"][0]
# If House or Villa------------------
if (convert_type == "H"):
      Type_h = 1
elif (convert_type == "V"):
      Type_h = 1
else:
      Type_h = 0

# If townhouse/unit or flat ---------
if (convert_type == "A"):
      Type_u = 1
elif (convert_type == "U"):
      Type_u = 1
elif (convert_type == "N"):
      Type_u = 1
elif (convert_type == "F"):
      Type_u = 1
else:
      Type_u = 0
```

```
Distance = float(features["suburb_distance_crime"][0])
Rooms = (float(features["bedrooms"]) - 1) / (12 - 1)
Bathrooms = (float(features["bathrooms"]) - 0 ) / (9 -
Cars = (float(features["cars"]) - 0) / (18 - 0)
Month = ((float(features["month"]) - 1) - 0) / (11 - 0)
Year = (float(features["year"])-2016)/(2024 - 2016)
```

Predicting

```
def predict_value(X):

    model = load_model()
    prediction = model.pred

    return prediction
```

```
X = pd.DataFrame([Rooms, Distance, Bathrooms, Cars, Landsize, Year, Month, Crime, Type_h, Type_t,
                  ['Rooms', 'Distance', 'Bathroom', 'Car', 'Landsize', 'Year', 'Month', 'Crime', '
model = load("model/xgboost_best_model_2024.joblib")
# run predict function from persist
predict = model.predict(X)[0]
# format value predicted
prediction_formated = f"{predict:,}"
```

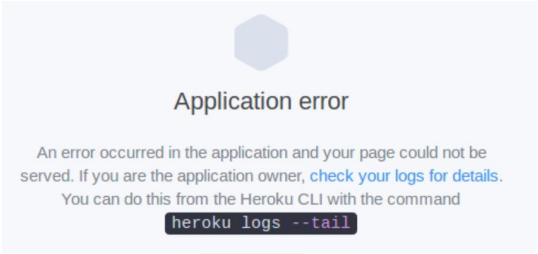
$1,652,283

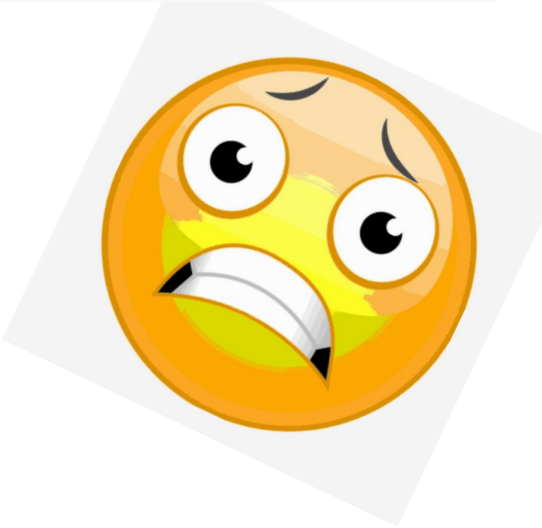
The predicted value for this pro
is made on 24/5/2021

Bedrooms: 2
athrooms: 1
Cars: 1
Landsize: 247m²
Property Type: House
Suburb: Carlton North
Suburb Distance to CBD: 3.2km

Heroku problems

- Using Flask and Selenium on localhost worked fine but when we deployed the app Heroku servers, errors popped up everywhere!!
  - Tried to run selenium with Heroku by installing additional packages
  - Unfortunately retrieved nothing using the logs and print statements to check
- Searched the internet (in panic with Hamim!!) for a solution
  - Found out domain.com.au blocked automated users and Heroku from accessing their site
  - Thought - “this app is a dead idea :(”





Solution: To bypass the detection of a bot we used requests library passing in a ‘header’ to make it look like we were a genuine user

```
headers = {
    'User-Agent': 'Chrome/5.0 (Macintosh; Mac OS X 10_11_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102'
html_content = requests.get(url, headers=headers)
```

- With this sorted we tried again but……

Heroku problems x2!!

- Our prediction would work on a localhost but not on Heroku!!
- Logs kept showing errors with xgboost not installing on Heroku servers even though it was the latest version
- After scouring the internet and many, many hours we found the solution.

Solution: Downgrading the version xgboost in the requirements.txt meant Heroku servers could install xgboost package and it fixed the problem””

```
requests==2.22.0
xgboost==1.4.2
splinter==0.12.0
```

And then…
It all worked!!

```
requests==2.22.0
xgboost==1.3.1
splinter==0.12.0
```

Actual re-enactment:

Demonstration

CHALLENGES

- Data availability:
  - The property data contained only suburbs in Melbourne. Therefore, the model does not work Victoria wide or on interstate properties.
  - The property data contained sales from 2016-2018, meaning our model wasn't trained on data that reflected the current housing market. This would limit the models accuracy to predict current prices.
- Modelling:
  - We originally created a multiple linear regression model as we believed this would suit our data, however after getting negative r2 values we had to explore different options.
- Data Scraping:
  - Domain blocks the use of Beautiful Soup and Selenium on their website, so once the app was deployed on Heroku this feature no longer worked. Changes then had to be made for the data scraping process.
  - The limitation of property features available on Domain, which in turn limited the features the model could be trained on.

POSSIBILITIES

With more time and resources we would like to explore:

- More recent data to make a more accurate model
- Extending the app to predict from other resources such as realestate.com
- The opportunity to compare multiple houses on the website

Q&A