SIT 307
DATA MINING AND MACHINE LEARNING
TRIMESTER-1 2021
GROUP PROJECT
EXPLORATORY DATA ANALYSIS

DATASET USED: MELBOURNE HOUSING DATASET
SUBMITTED TO: SOMAIYEH MAHMOUDZADEH

Name	***Student ID***	***Contribution***
Navdeep Singh Randhawa	219249449	5: Coding and report writing.
Gurpreet Singh	218701668	5: Report writing and presentation.
Abhishek Kapoor	218590431	5: Coding and report writing.
Michael Rai	218680443	5: Coding and report writing.
Iftekhar Qureshi	218676618	5: Coding and report writing.
Rawan Awadh K Alharbi	219113465	5: Presentation and Report writing.

Section 1: Introduction and getting to know your data.

WHY?

Melbourne Housing Market data was collected to analyse house prices in the city of Melbourne, Australia. The dataset contains information on the trend of house prices over the last few decades in Melbourne. The dataset shows which parts of the city are most expensive. Additionally, there are some extra information about the houses and their features.

By exploring the dataset, we can find valuable insights to predict the next big trend in the Melbourne housing market. The insights can help real estate businesses and future house owners. Investment in housings might be more profitable in some places. For example, apartments in Melbourne CBD might be worth buying while townhouses are better investments in the suburbs.

HOW?

Our dataset is a snapshot of Melbourne Housing Market dataset found on kaggle. The data was scraped into a spreadsheet from publicly available results posted every week on Domain.com.au. The data was not entirely cleaned which is why cleaning is required before any analysis is done.

Initial Observation and Hypothesis:

The hypothesis being tested here in this project is that the price of the houses depends strongly on certain features, which namely are distance from the CBD, age of the house, number of rooms, bathrooms, car spaces and council areas. To test this hypothesis, data had to be cleaned and later analysed using relevant techniques. For example, there are some missing values in the columns, there was expected relationship (correlation) between certain variables such as price and rooms, which can be verified through exploratory analysis.

There are several variables in our chosen dataset as follow:
All the variables have their own importance that they provide to the dataset and one can get full information about the data.

Variable Name	Description	Data Type
1. Address	Address of the property	String
2. Suburb	Name of the suburb	String
3. Distance	Distance from CBD in Km's	Float
4. Council Area	Governing council for the area	String
5. Price	Price in dollars	Float
6. Year Built	Building year of the house	Float
7. Land Size	Total Land used	Float
8. Building Area	Actual Building area of the construction	Float

9. Car	Number of Parking spaces.	Float
10. Bedroom	Number of bedrooms	Float
11. Bathroom	Number of bathrooms	Float
12. Property Count	Number of properties that exist in that particular suburb	Float
13. Region name	North , South , West , East	String
14. Date	Date on which the house was sold	String
15. SellerG	Name of Real Estate Agent	String
16. Method	The selling state of the property	String

Section 2: Exploratory Data Analysis and Results

DATA CLEANING:

Data cleaning is the process of identifying data that is incorrect and then fixing the incorrect data to replace or delete them with correct data accordingly. Incorrect data will produce inaccurate results. Inaccurate results can greatly harm businesses that rely on data analysis. We did extensive cleaning for our dataset. Methods we used for cleaning and their details are given below-

- **Missing Values:**

We found 4 columns containing null values: Car, BuildingArea, YearBuilt and CouncilArea. There can be multiple approaches to deal with missing values. (Brownlee, 2020)

Methods to handle missing value:

```
In [691]: df.isnull().sum()
Out[691]: Suburb            0
          Address           0
          Rooms             0
          Type              0
          Price             0
          Method            0
          SellerG           0
          Date              0
          Distance          0
          Postcode          0
          Bathroom          0
          Car              60
          Landsize          0
          BuildingArea   6332
          YearBuilt      5268
          CouncilArea    1332
          Lattitude         0
          Longtitude        0
          Regionname        0
          Propertycount     0
          Count             0
          dtype: int64
```

1. Dropping the feature:

The feature BuildingArea was dropped because it contained the highest number of null values, which was about 40 percent of total data records. Other than that, we calculated the correlation of BuildingArea with price of the house.

2. Linear Regression Imputation:

Linear regression model was utilised to predict null values in the feature YearBuilt. The data used to build the model was isolated and had no null values for any of the concerned features. The focus of study was YearBuilt, hence YearBuilt was the response variable while Distance was the explanatory variable. After calculation of the coefficient and slope of the equation, values were used to fill the null values.

3. Imputation with Mean:

Car feature had very a smaller number of null values, thus it was feasible to fill up these values with the mean calculated for all the data records.

4. Handling String data:

There were only a small number of missing data in the CouncilArea feature. To fix that data, we use the provided data and replace the null values by using simple python algorithm.

- **Outliers:**

First step to finding outliers was checking our data using histogram to see if our data was skewed. Yearbuilt was slightly left skewed (negatively skewed). We decided to calculate the z-score to find the outliers using that column. There were 6 such rows 3 std deviations below the mean (high z-score) which had to be discarded. (Sharma, 2018)

```
df['z_yearbuilt']=(df.YearBuilt-df.YearBuilt.mean())/df.YearBuilt.std()
```

```
df[df['z_yearbuilt']<-3]
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Landsize
2079	Collingwood	2/79 Oxford St	2	u	855000.0	S	Nelson	3/09/2016	1.6	3066.0	...	2886.0
2554	Fitzroy	11 Henry St	2	h	677000.0	S	Chambers	27/11/2016	1.6	3065.0	...	67.0
4843	Prahran	602/220 Commercial Rd	2	u	841000.0	S	hockingstuart	18/03/2017	4.5	3181.0	...	0.0
5405	Richmond	22a Stanley St	3	h	1600000.0	S	Biggin	24/09/2016	2.6	3121.0	...	80.0
5860	St Kilda	51/167 Fitzroy St	3	u	1600000.0	PI	Kay	25/02/2017	6.1	3182.0	...	0.0
9968	Mount Waverley	5 Armstrong St	3	h	1200000.0	VB	McGrath	24/06/2017	14.2	3149.0	...	807.0

6 rows × 23 columns

```
df[df['z_yearbuilt']>3]
```

Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Landsize	BuildingArea	YearBuilt	C

0 rows × 23 columns

```
df=df.drop(2079)
df=df.drop(2554)
df=df.drop(4843)
df=df.drop(5405)
df=df.drop(5860)
df=df.drop(9968)
df.info()
```

- **Handling Duplicate values:**

It was possible to have duplicated values in all the columns except Address, hence it was necessary to investigate about such values. Initial exploration using unique() hinted that the feature could have duplicated values, as the output of this method differed from total count of data records. The duplicated() methods was called for the column and there were 197 such rows containing duplicate addresses which were dropped. (Rachuta, 2021)

```
duplicate = df[df['Address'].duplicated()]
duplicate
```

	Suburb	Address	Rooms	Type	Price	Method	SellerG	Date	Distance	Postcode	...	Car	Landsiz
40	Airport West	50 Bedford St	3	h	770000.0	SP	Nelson	4/03/2017	13.5	3042.0	...	1.0	0.
197	Altona North	21 Hatherley Gr	3	h	700000.0	VB	Jas	10/12/2016	11.1	3025.0	...	3.0	554.
575	Balwyn	112 Belmore Rd	5	h	3020000.0	PI	Jellis	28/05/2016	9.7	3103.0	...	2.0	715.
615	Balwyn North	3 Clive Ct	4	h	2130000.0	PI	RW	8/10/2016	9.2	3104.0	...	2.0	1274.
667	Balwyn North	41 Helston St	4	h	1900000.0	VB	One	22/08/2016	9.2	3104.0	...	4.0	587.
...	...	...	...	...	...	...	...	...	...	...	...	...	...
13371	Brighton East	375 South Rd	5	h	1650000.0	SP	Hodges	26/08/2017	10.3	3187.0	...	5.0	773.
13383	Burwood	23 Cromwell St	3	h	1238000.0	S	Buxton	26/08/2017	10.4	3125.0	...	1.0	735.
13421	Footscray	39 Moore St	3	h	755000.0	PI	hockingstuart	26/08/2017	5.1	3011.0	...	2.0	334.
13429	Frankston South	3 Diosma Ct	3	h	1155000.0	S	hockingstuart	26/08/2017	38.0	3199.0	...	2.0	2405.
13444	Hampton East	3 Besant St	3	h	1280000.0	SP	Buxton	26/08/2017	13.8	3188.0	...	3.0	658.

197 rows × 22 columns

```
df=df.drop_duplicates(subset=['Address'])
df.info()
```

- **Unnecessary Features:**

There were two columns containing similar information- ‘Rooms’ and ‘Bedroom2’ both containing information on the number of rooms a home contains. After visualising, we did not get much difference between them. Hence, it was more favourable to drop bedroom2 column for better data analysis. (Paul, 2020)

```
# Examine Rooms v Bedroom2
df['Rooms v Bedroom2'] = df['Rooms'] - df['Bedroom2']
df
```

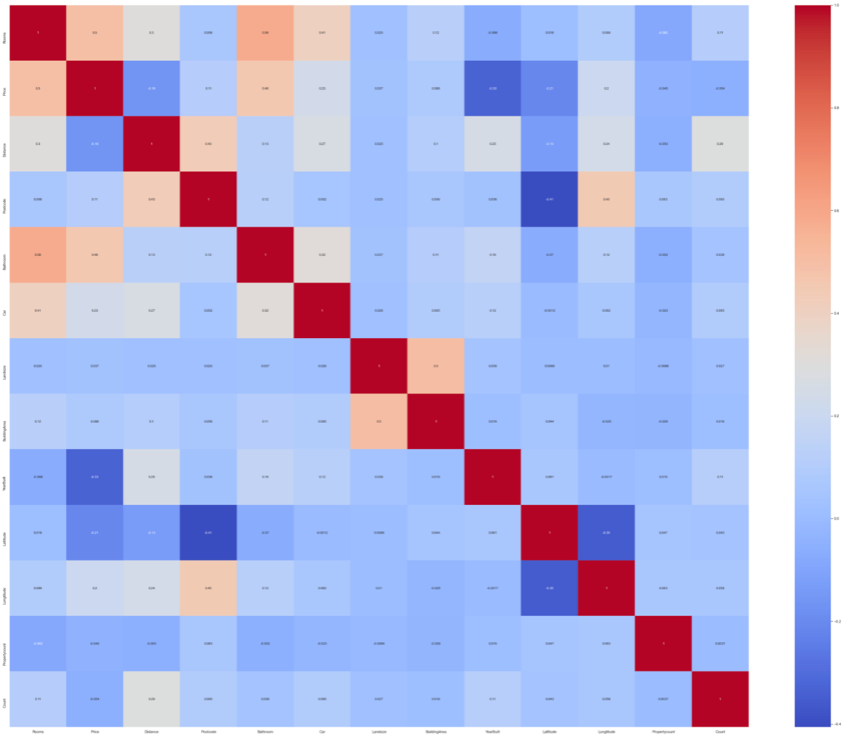
CORRELATION:

Correlation is term that is used to denote some form of association among the features in the dataset in real life, for example there could be some association between price and number of rooms in the Melbourne housing dataset. The association defined is linear, which means if one features value increases or decreases, other features value also increases or decreases depending on the type of association (Hayes, 2021).

Correlation patterns for dataset:

Heatmap of the correlation matrix:

```
plt.figure(figsize=(50,40))
ax = sns.heatmap(corr, annot=True,cmap="coolwarm")
```



Price of houses vary depending on features included in the house. As we see in our daily life, big houses or houses with high number of bedrooms and bathrooms are particularly expensive. This was slightly evident when we calculated the correlation between the variables and found out price and rooms have a correlation of 0.53 (moderate correlation) and price and bathroom have a correlation of 0.49.

Surprisingly, we found expected negative correlation between price and Distance but this correlation is extremely weak (correlation of -0.15). This shows that even though we might think the further the distance from CBD, the lower should be prices of the houses, this is not entirely true.

INITIAL HYPOTHESIS TESTING:

Hypothesis testing refers to take a guess about something in form of a statement and then test it using statistical means by calculating p value and testing against the alpha value. Here in the dataset, as the main feature of interest is the price of house, the hypothesis is created for the feature, which refers to one sample hypothesis testing, in which the values are tested for one feature against themselves. (Majaski, 2020)

The hypotheses that were tested are as follow:

1) The mean price of the houses built after 1990 is same as the overall mean price, this hypothesis was rejected based on the p value and alpha value set to 5 percent.

```
null_hypothesis="NULL HYPOTHESIS: The mean price of houses built after 1990 has no difference"
new_houses = df.loc[(df['YearBuilt']>1990)]
_,p_value=stats.ttest_1samp(a=new_houses['Price'],popmean=df['Price'].mean())
if(p_value<0.05):
    print(null_hypothesis)
    print("P Value: ",p_value)
    print("HYPOTHESIS REJECTED")
else:
    print(null_hypothesis)
    print("HYPOTHESIS ACCEPTED")
```

```
NULL HYPOTHESIS: The mean price of houses built after 1990 has no difference
P Value:  6.242325340339285e-23
HYPOTHESIS REJECTED
```

2) The mean price of houses in Council area Yarra is equivalent to overall mean price of the houses, this hypothesis was also rejected based on the p value and the alpha set to 5 percent.

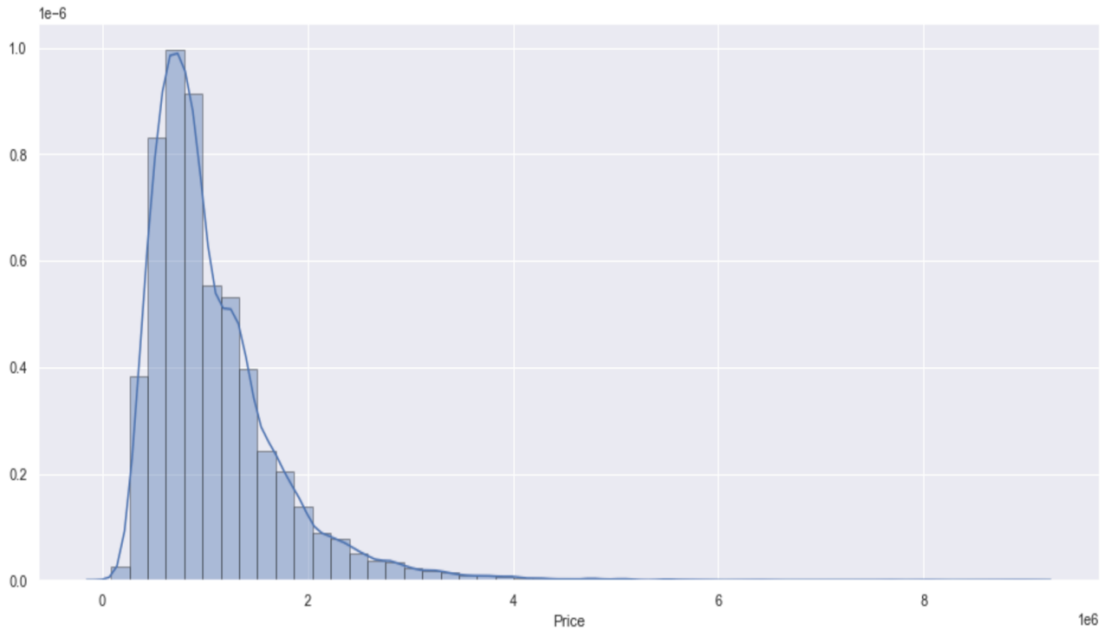
```
null_hypothesis="NULL HYPOTHESIS: The mean price of houses in Coucil area Yarra is equivalent
to overall mean price. "
new_houses = df.loc[(df['CouncilArea']=='Yarra')]
_,p_value=stats.ttest_1samp(a=new_houses['Price'],popmean=df['Price'].mean())
if(p_value<0.05):
    print(null_hypothesis)
    print("P Value: ",p_value)
    print("HYPOTHESIS REJECTED")
else:
    print(null_hypothesis)
    print("HYPOTHESIS ACCEPTED")
```

```
NULL HYPOTHESIS: The mean price of houses in Coucil area Yarra is equivalent to overall mean
price.
P Value:  0.015886577336741415
HYPOTHESIS REJECTED
```

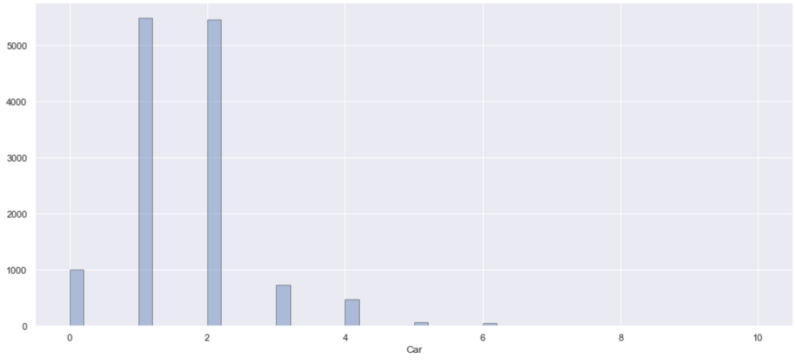
KEY FEATURES, VISUALISATION AND OVERALL HYPOTHESIS TESTING:

The features of the dataset were concluded as the result of visualisations in form of graphs.

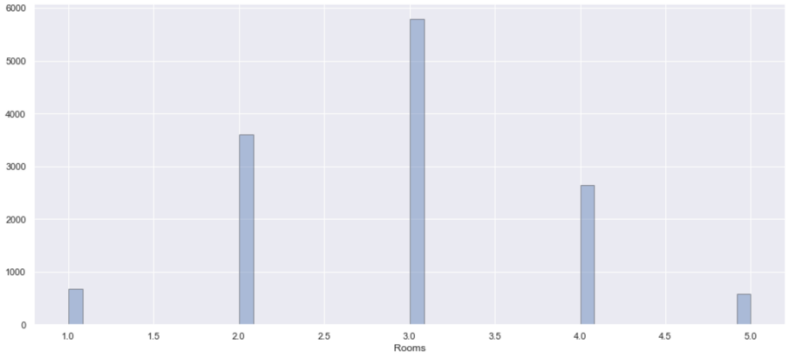
1) The prices of the houses are mostly in range of $100,000 to $150,000.



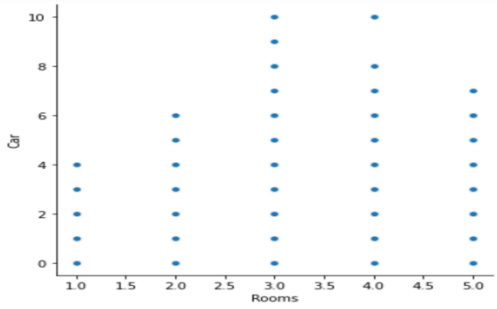
2) The majority of houses either have 1 car or 2 cars across Melbourne.



3) The majority of houses have either 1 bathroom or 2 bathrooms and either 2,3 or 4 rooms across Melbourne.

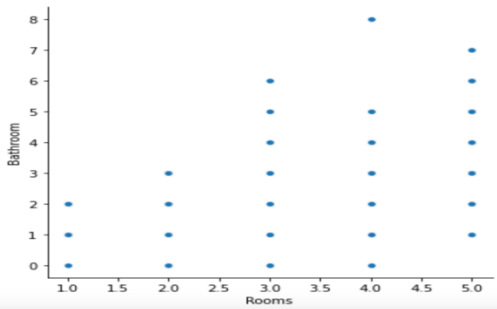


4) More the number of rooms, more the number of bathrooms and cars.

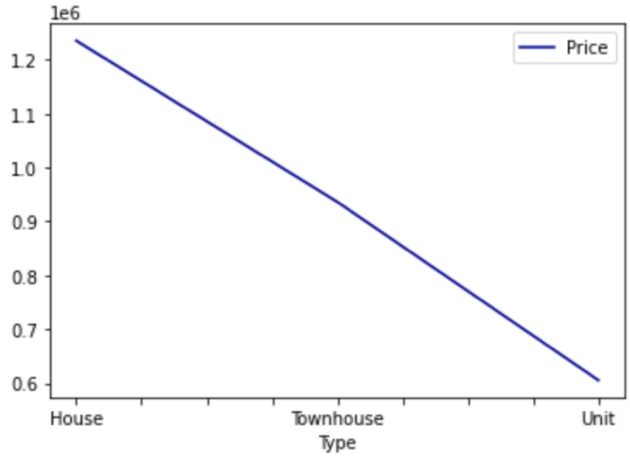



```
#rooms vs bathroom
sns.relplot(x="Rooms",y="Bathroom",data=df)
```

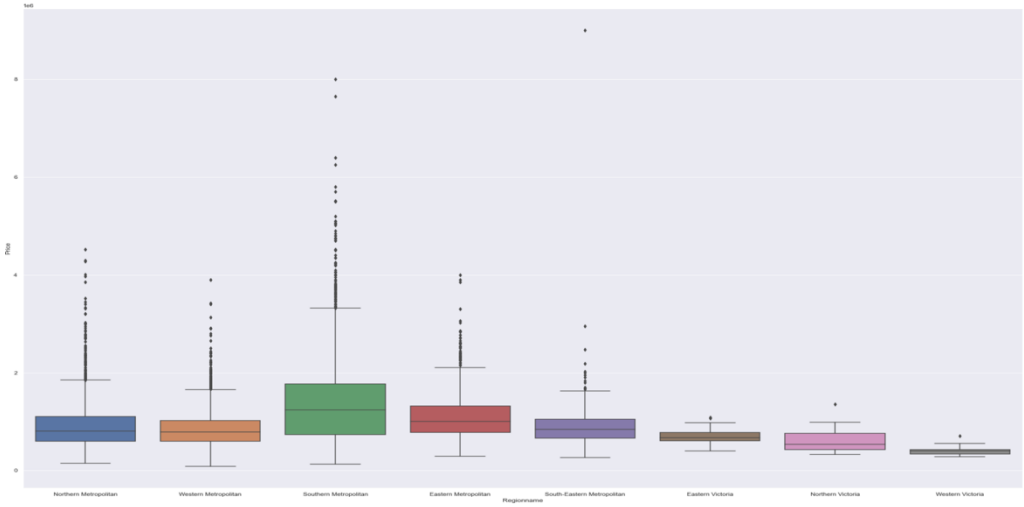
```
<seaborn.axisgrid.FacetGrid at 0x7fb188c94eb0>
```



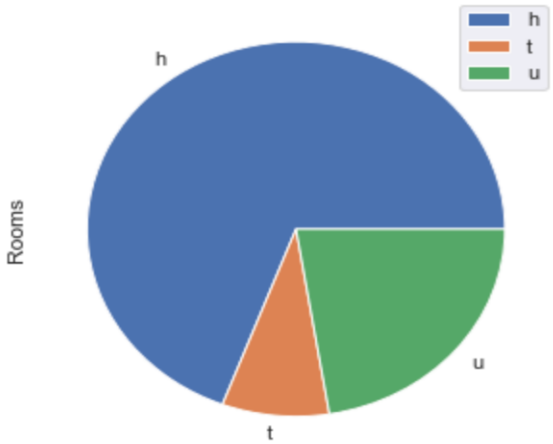
5) House category has highest mean for the prices, followed by townhouses and units.



6) Southern Melbourne has highest average house price whereas Western Victoria has lowest.



7) Most of the houses are categorised as type ‘house’ followed by ‘units’ and ‘townhouses’.



Section 3: Conclusion

We chose the Melbourne housing dataset as it contains a considerable amount of data from which we can work with. The primary attention, when working with dataset, should be given to the dataset itself. Data preparation is the main concept where one should be careful while corking with the dataset. Because wrong data can lead to wrong conclusions and wrong results. We tried to find inaccuracies in the dataset, and we came to know many such as there were duplication of data, null values for data, and wrong data. We used multiple methods for cleaning the data such as deleting some part of it, linear regression and basic python commands.

After cleaning the data, visualizations were made. Multiple visualizations were made among different features such as price vs distance, price vs bedroom, etc. Different forms of visualizations were also used such as scatterplot, boxplot, etc to get a clearer idea of the dataset.

Another part of the project was to analyse hypothesis. Different hypotheses were analysed to get a clearer insight into the dataset. We had number of hypotheses such as the price of the house against the distance from the city or the number of rooms. Interestingly the only half of our hypothesis was true i.e., the price is strongly depending on the number of rooms in the house whereas the price is not strongly dependent on the distance of the house from the city which is analysed in the jupyter notebook.

For the next part of the project, multiple machine learning models will be built to make some prediction models. These prediction model will calculate the price of a particular house by using multiple features such as number of rooms etc.

Section 4: References

- Hayes, A., 2021. *Correlation*. [online] Investopedia. Available at: <https://www.investopedia.com/terms/c/correlation.asp> [Accessed 22 April 2021].
- Majaski, C., 2020. *How Hypothesis Testing Works*. [online] Investopedia. Available at: <https://www.investopedia.com/terms/h/hypothesistesting.asp> [Accessed 22 April 2021].
- Brownlee, J., 2020. *How to Handle Missing Data with Python*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/handle-missing-data-python/> [Accessed 22 April 2021].
- Rachuta, K., 2021. *Dealing with duplicates in pandas DataFrame*. [online] Medium. Available at: <https://medium.com/@kasiarachuta/dealing-with-duplicates-in-pandas-dataframe-789894a28911> [Accessed 22 April 2021].
- Sharma, N., 2018. *Ways to Detect and Remove the Outliers*. [online] Medium. Available at: <https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba> [Accessed 22 April 2021].
- Paul, S., 2020. *Beginner's Guide to Feature Selection in Python*. [online] DataCamp. Available at: <https://www.datacamp.com/community/tutorials/feature-selection-python> [Accessed 22 April 2021].