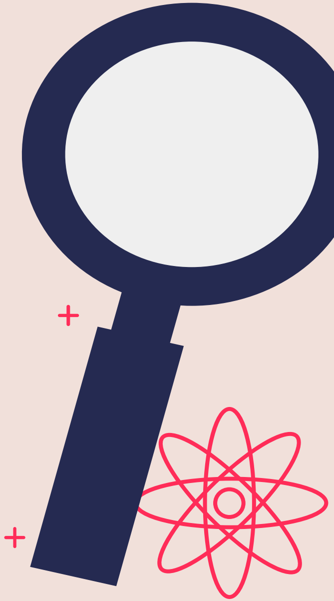
DATA SCIENCE & ARTIFICIAL INTELLIGENCE

Seattle Airbnb Open Dataset

PROBLEM STATEMENT

Helping owners to predict appropriate prices to determine if their house is undervalued or overvalued, and also performing sentiment classification to find out how to improve their houses from reviews.

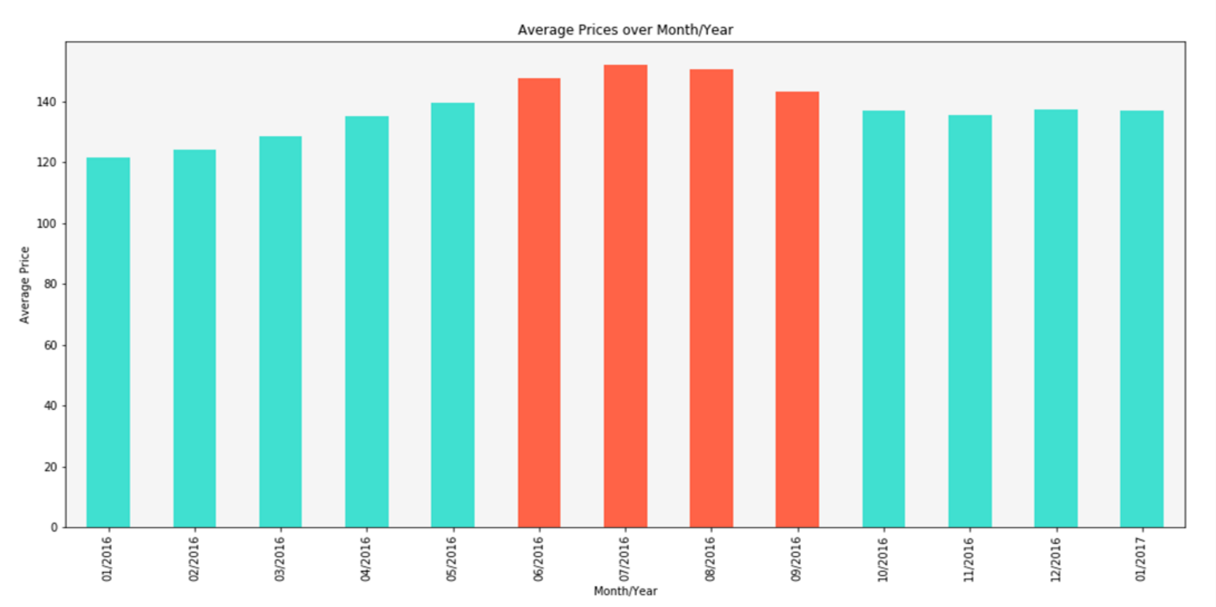


EXPLORATORY DATA ANALYSIS

Initial investigation on data to visualise patterns

PRICING TREND OVER A YEAR

Interesting findings

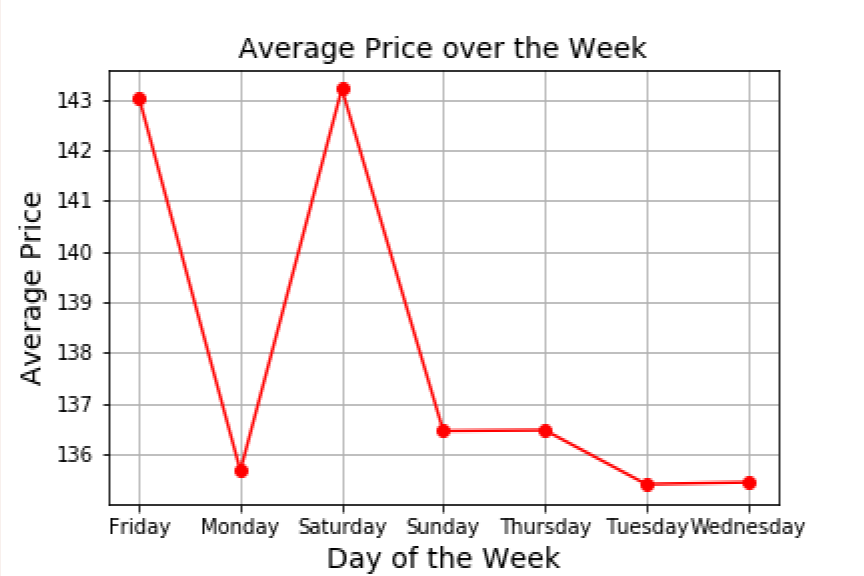


Observation: Prices peaked during the months of June to September

Possible explanation: Summer break period where more people go on holidays

PRICING TREND OVER A WEEK

Interesting findings

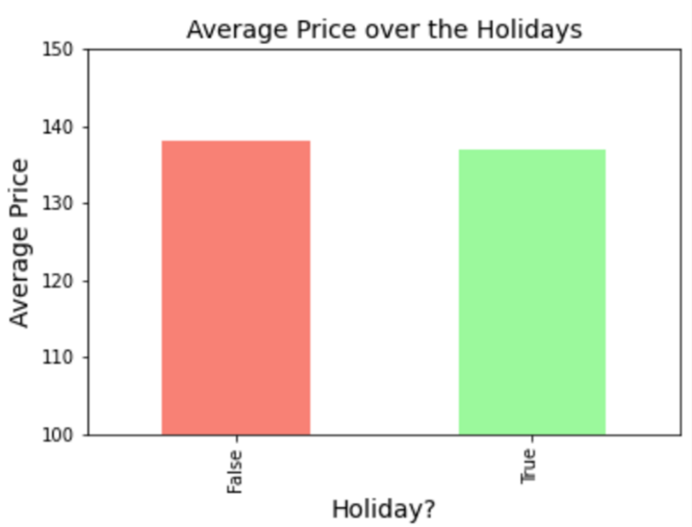


Observation: Fridays and Saturdays have significantly higher prices

Possible explanation: Weekend time frame, where most people take a break from work, thus more people might be renting Airbnbs.

PRICING TREND DURING HOLIDAYS

Interesting findings



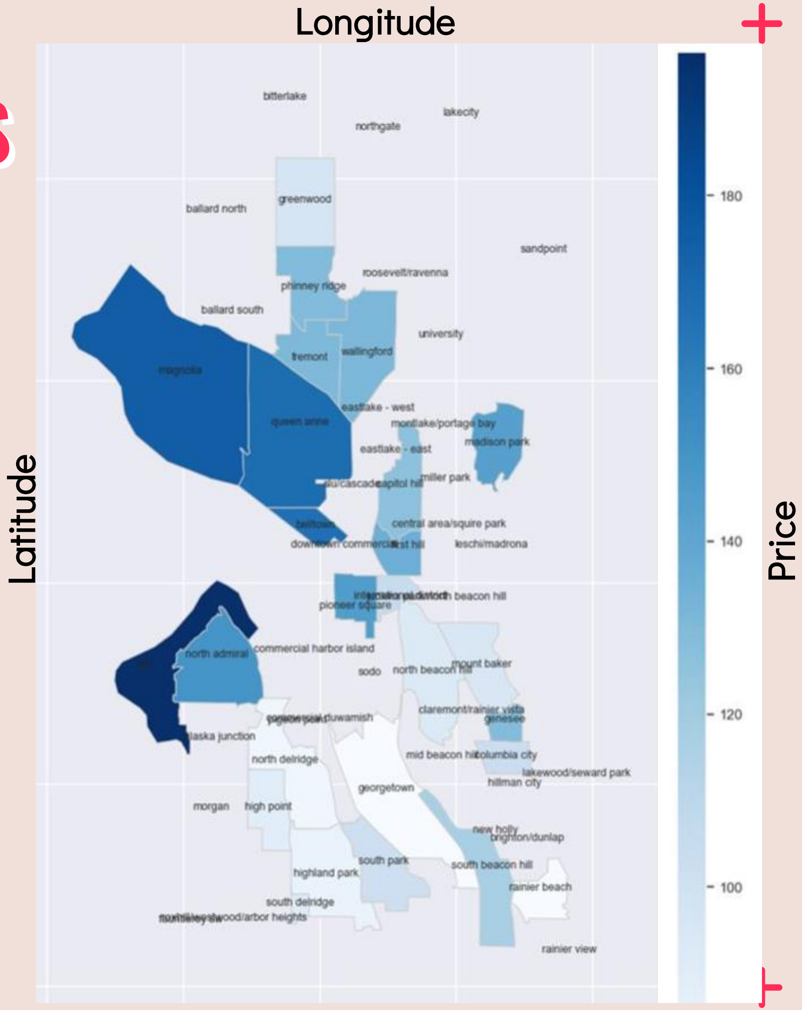
Observation: Rather equal in pricing for both holidays and non-holidays which was surprising to us

EXPLORATORY DATA ANALYSIS

Interesting findings: **Pricing with location**

Observation: Prices are highest around the central area of Seattle

Possible explanation: Most of the attractions are located there hence, there is a higher demand for houses there



MACHINE LEARNING MODELS WE'VE USED

Linear Regression

Predicting Price through uni and multi-variable regressions

Sentiment Analysis

Sieving through common complaints for improvement of homes

Random Forest Regression

Predicting Price through uni-variable regression

Pycaret

Comparing different machine learning models and determining the best one

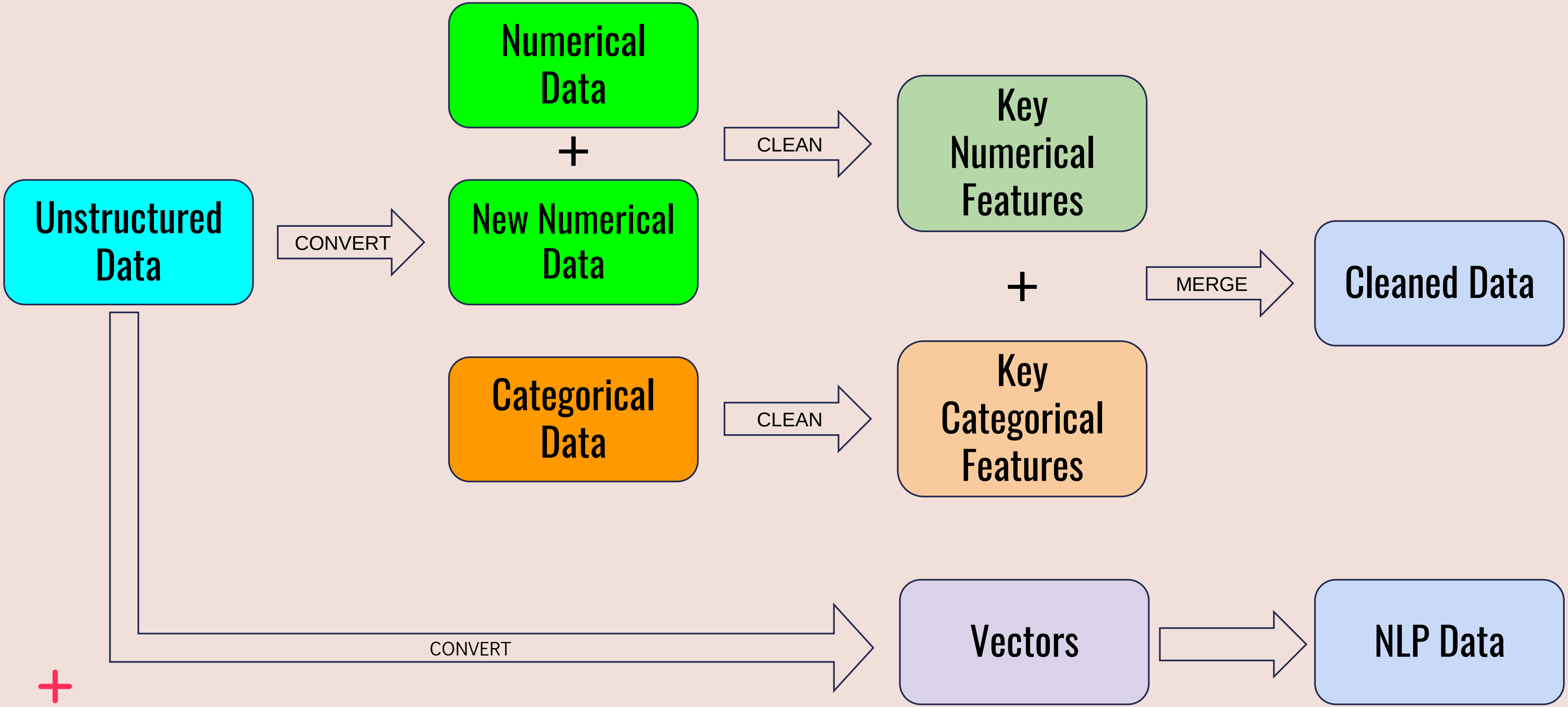
Keras

Construction of neural network

DATA PREPARATION

Cleaning and structuring the data

DATA CLEANING & STRUCTURING



DATA CLEANING & STRUCTURING

01. CONVERTING UNSTRUCTURED DATA INTO NUMERICAL FORM

Converting unstructured data types into numerical form for implementation to regression models (price, transit, host_verfications)

```
Name: price, Length: 3818, dtype: object
```

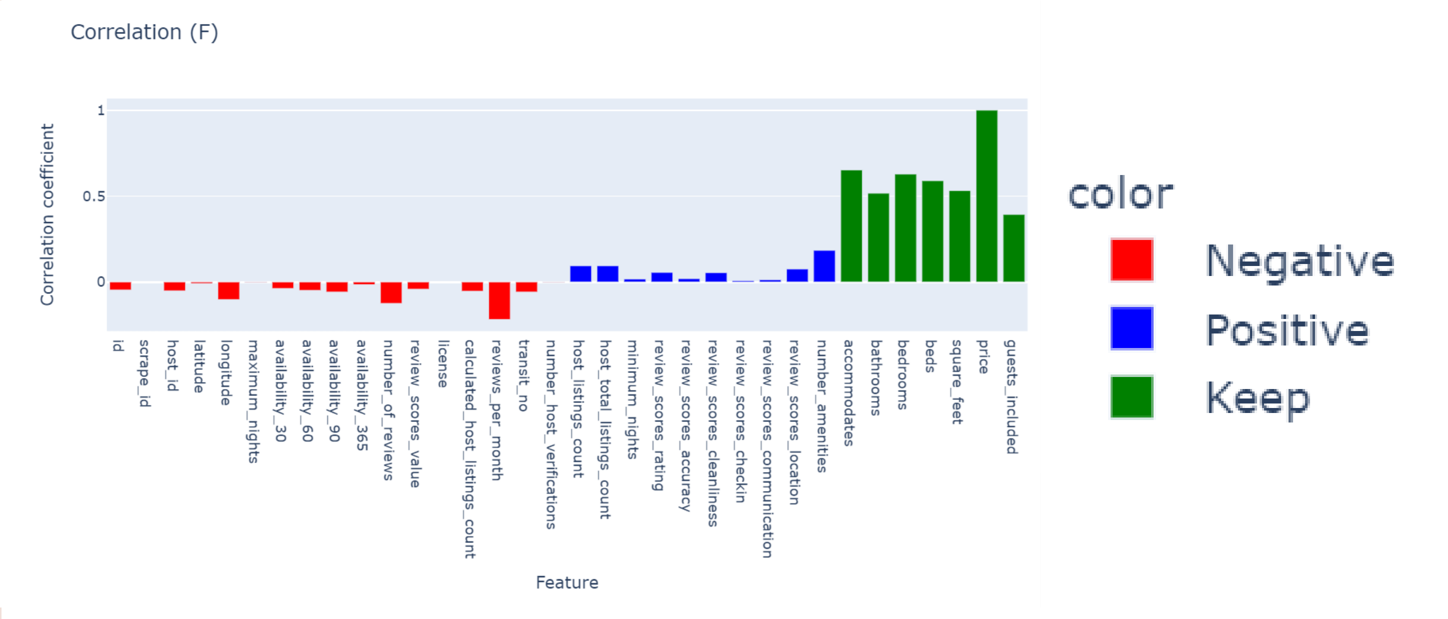
→ int64

transit	→	transit_no
NaN		0
Convenient bus stops are just down the block, ...		2
A bus stop is just 2 blocks away. Easy bus a...		1
NaN		0
The nearest public transit bus (D Line) is 2 b...		1

DATA CLEANING & STRUCTURING

02. KEEPING NUMERICAL FEATURES THAT HAVE STRONG CORRELATION

Only correlations with absolute value greater than 0.35 are kept while others are dropped

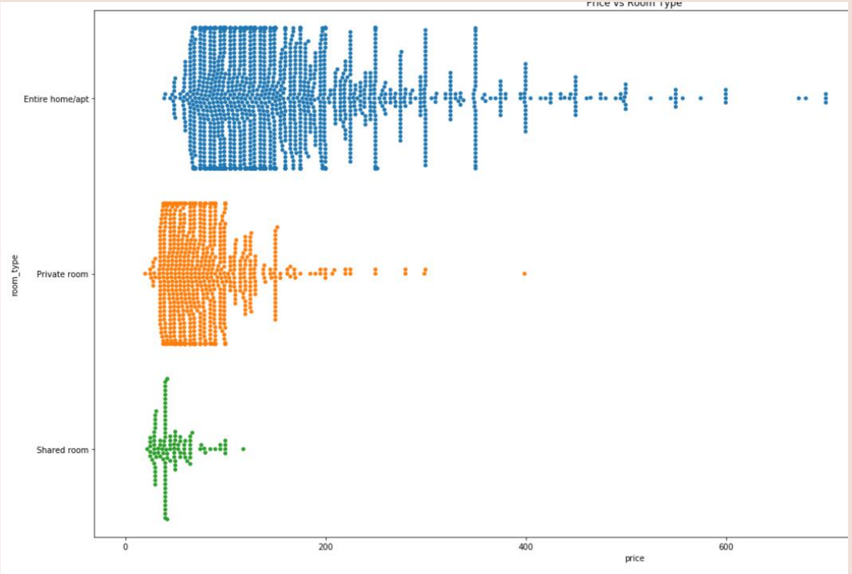


DATA CLEANING & STRUCTURING

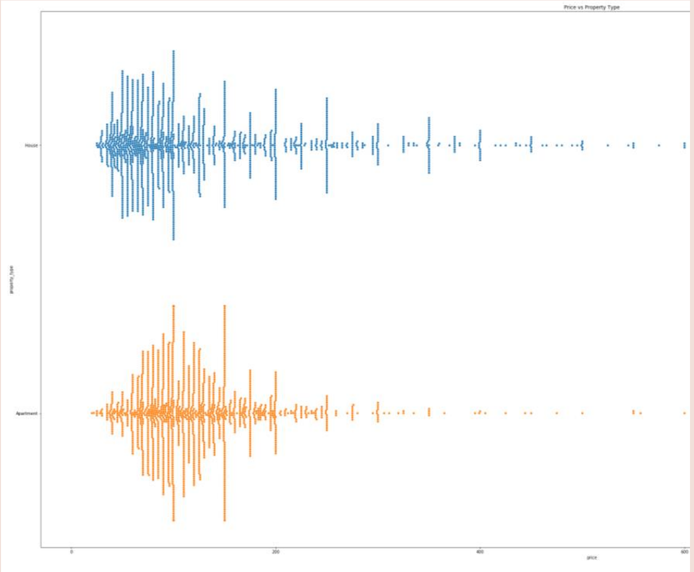
03. KEEPING CATEGORICAL FEATURES THAT DISTINGUISH PRICE

Exploring categorical data and deducing if the feature aids in distinguishing prices

Price vs Room Type



Price vs Property Type



DATA CLEANING & STRUCTURING

04. MERGING DESIRED FEATURES TOGETHER

We merged the desired categorical and numerical features into a single dataframe and save it as a csv for input into machine learning notebook.

Numerical Data + Categorical Data

id	property_type	room_type	neighbourhood	price	accommodates	bathrooms	bedrooms	beds	guests_included
241032	Apartment	Entire home/apt	Queen Anne	85	4	1.0	1.0	1.0	2
953595	Apartment	Entire home/apt	Queen Anne	150	4	1.0	1.0	1.0	1
3308979	House	Entire home/apt	Queen Anne	975	11	4.5	5.0	7.0	10
7421966	Apartment	Entire home/apt	Queen Anne	100	3	1.0	0.0	2.0	1
278830	House	Entire home/apt	Queen Anne	450	6	2.0	3.0	3.0	6

→ *cleaned_listing.csv*

DATA CLEANING & STRUCTURING

05. CONVERTING WORDS TO TOKENS (for sentiment analysis)

STEP 01

Words
“The cat sat on the mat.”

Tokens
“The”, “cat”, “sat”, “on”, “the”,
“mat”

Removing punctuations and converting the words to tokens.

STEP 02

Tokens
“~~The~~”, “cat”, “sat”, “~~on~~”, “~~the~~”,
“mat”

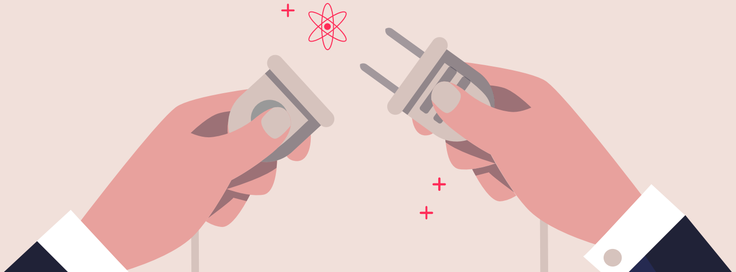
Removing stopwords

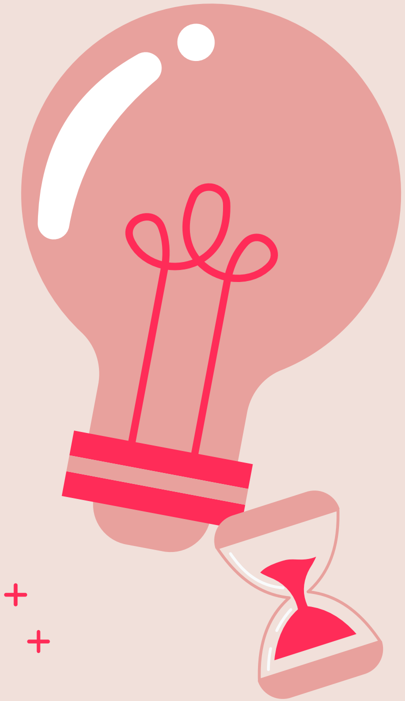
STEP 03

	original_word	lemmatized_word
0	trouble	trouble
1	troubling	trouble
2	troubled	trouble
3	troubles	trouble

	original_word	lemmatized_word
0	goose	goose
1	geese	goose

Lemmatize Words





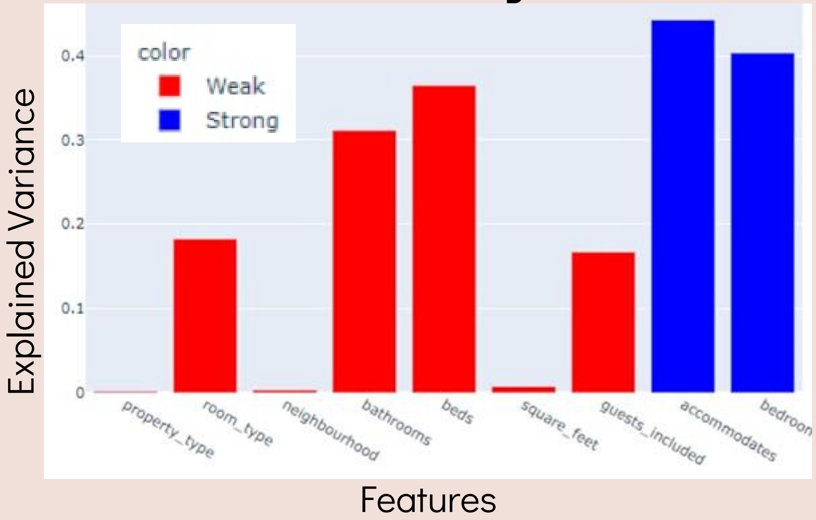
MACHINE LEARNING

Tools and techniques to analyse the data

LINEAR REGRESSION on UNI-VARIABLES

Square Feet Linear Regression Actual vs Result

Correlation for each single variable



Uni Variables are not an effective predictor of price

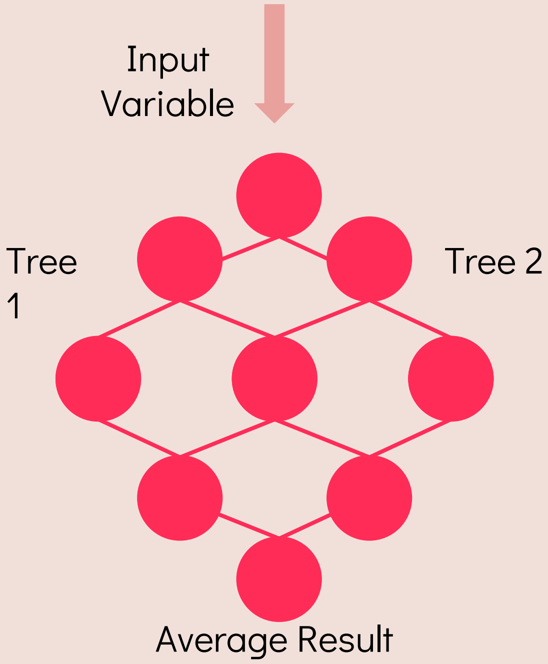
LINEAR REGRESSION on MULTI-VARIABLES

Multivariable Linear Regression Actual vs Result

Since more points were on the best fit line, multi variable linear regression is an effective predictor of price.

RANDOM FOREST REGRESSION on UNI-VARIABLES

Similar to decision trees, Random forest regression uses the ensemble method which creates multiple models and combines them to improve results



Mean Squared Error obtained from uni-variables

property_type 8456.256473742025	accommodates 4729.695370463768
room_type 6884.199908146492	bathrooms 5673.0327102035335
neighbourhood 7940.975906691937	bedrooms 4621.51516299114

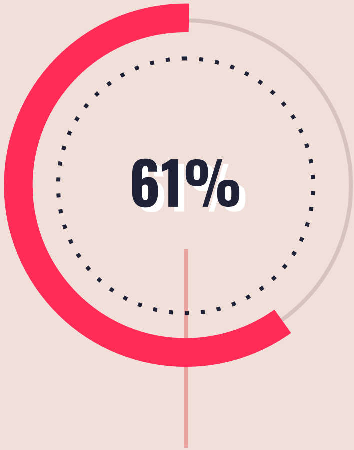
PYCARET

Pycaret is a Python low-code library that **helps you perform model selection** allowing us to spend less time coding and more time on results and data analysis.

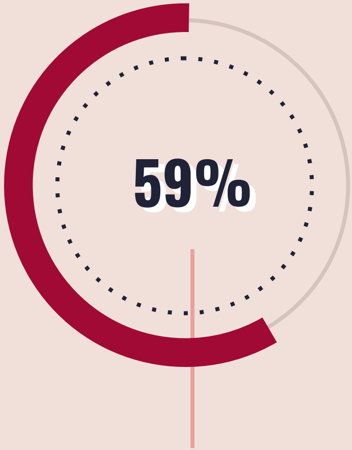
	Model	MSE	RMSE	R2
gbr	Gradient Boosting Regressor	2420.8161	49.0773	0.6072
ghtgbm	Light Gradient Boosting Machine	2565.2213	50.5277	0.5830
br	Bayesian Ridge	2620.0793	51.0777	0.5747
ridge	Ridge Regression	2642.1222	51.2947	0.5711
lr	Linear Regression	2669.4086	51.5665	0.5666

After comparing ML models using PYCARET, **gradient boosting regressor** has the highest R^2 value, hence it is the best model for predicting price.

GRADIENT BOOSTING REGRESSOR (after tuning) IN PREDICTING PRICES



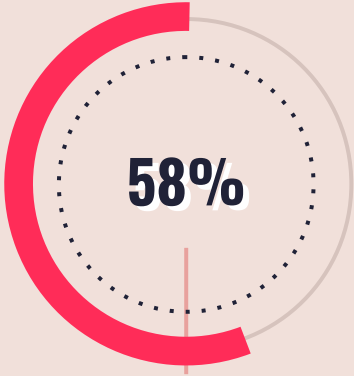
TRAIN DATASET



TEST DATASET

As seen from the results, the model performed a little worse on the test dataset, which is as expected.

NEURAL NETWORK via KERAS



TEST DATASET

Mean Square Error : 3197

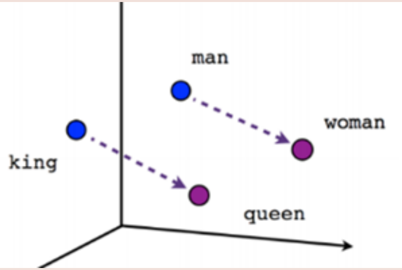
Multiple layers

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_51 (Dense)             (None, 128)               1280
_________________________________________________________________
dense_52 (Dense)             (None, 256)               33024
_________________________________________________________________
dropout_8 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_53 (Dense)             (None, 256)               65792
_________________________________________________________________
dense_54 (Dense)             (None, 128)               32896
_________________________________________________________________
dropout_9 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_55 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_56 (Dense)             (None, 1)                 65
=================================================================
Total params: 141,313
Trainable params: 141,313
Non-trainable params: 0
_________________________________________________________________
```

NATURAL LANGUAGE PROCESSING

Where machines **decipher and understand** human language

UNSUPERVISED SENTIMENT ANALYSIS



Negative words

```
'wifi_unstable', 0.99560749530
'biggest_complaint', 0.99486327
'old_nasty', 0.99435448646545
```

Positive words

```
lifetime_experience', 0.9917296171
georgeous', 0.9916488528251648),
unbelievable_hospitality', 0.99162
```

sentence	prediction
e perfect location everything	1
om central location beautiful bu...	1
ent great neighborhood kind apa...	1

01

Converted words into vectors using *Word2Vec*

(similar words are close together)

02

Separating words into positive and negative groups using *k-means clustering*

03

Assign weights to words within a sentence using *tf-idf vectorizer*

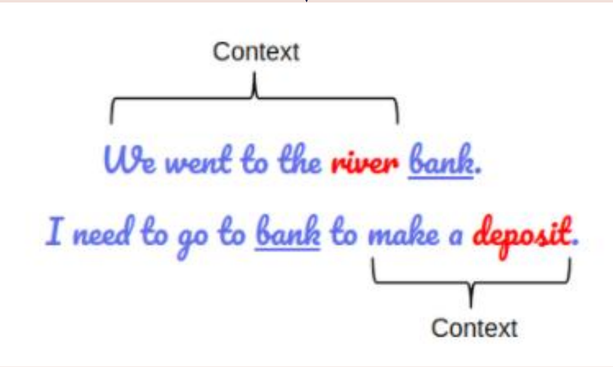
04

Aggregated sentiment and tf-idf scores at sentence level and output prediction (0 for negative, 1 for positive)

SUPERVISED SENTIMENT ANALYSIS

BERT

Bidirectional Encoder Representations from Transformers



Masking
Hides words from the model to better predict next words within a sentence

Attention
Enable the model to have longer memory and retains context from previous words

SUPERVISED SENTIMENT ANALYSIS

Input (2000 labelled sentences)

↓

Encoding (token, mask, segment)

↓

BERT model (512 tokens)

Understand sentences

↓

Keras layers

- Dense (64)
- Dropout
- Dense (32)
- Dropout

Classify sentences

↓

Output (3)

→

-1: negative
0: neutral
1: positive

SUPERVISED SENTIMENT ANALYSIS

Accuracy on train set: **93%**

Accuracy on test set: **89.5%**

<u>Positive word cloud generated</u>

<u>Negative word cloud generated</u>

Conclusion

- Outcomes
- Reviewing objectives
- Work Allocation

MACHINE LEARNING OUTCOME

PRICE (REGRESSION)

Multivariate Gradient Boosting Regressor

	Train	Test
MSE	2410	2540
R^2	0.608	0.598

SENTIMENT (CLASSIFICATION)

BERT with neural network classifier

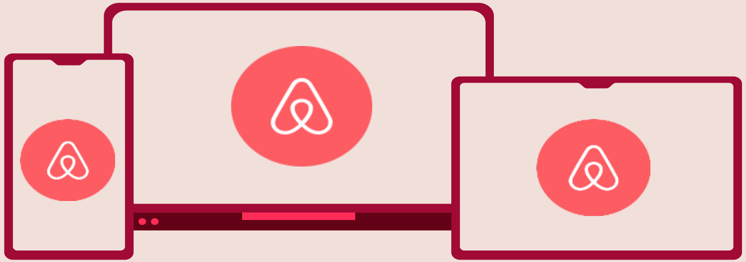
	Train	Test
Accuracy	93.6%	89.5%

REVIEWING OBJECTIVES

Predicted prices of houses using regression models	✓
Highlighted improvements that can be made by owners using Sentiment Analysis	✓
Gained interesting insights on the data from EDA	✓
Experimented with various machine learning tools outside of syllabus	✓
Gained knowledge on Natural Language Processing	✓

FUTURE IMPLEMENTATIONS

- Create an application for AirBnB hosts
- Hosts receive advice on house pricings based on features
- Hosts receive notifications when a negative review is given



WORK ALLOCATION

Leck Boon Kong

Machine Learning and Sentiment Analysis

Marcus Sing

Cleaning of the Dataset and Machine Learning

Samuel Ang

Exploration and Cleaning of the Dataset

Rei Ong

Exploration and Visualisation of the Dataset

Thank You!

Any Questions?