
# Group 5

Pratyush Kumar Pandey, Ashton Lim, Heather Chew, Gao Sitian

# Problem Formulation

**Statement:** Are review ratings of the listings reflective of its' features?

**Aim:** Create the best model to predict ratings of the listings and find the features that truly reflect the ratings

**Response variables:**

1) *review_scores_rating* - provided in the dataset
2) *analyser_review_rating* - derived using sentiment analysis on texts in reviews

# Practical Motivation

## Hotel review snippets

Some hotels have review summaries licensed from TrustYou, a third party. TrustYou creates review summaries and aggregates scores using reviews from across the web.

### Review summary

Write a review

**Rooms** · 4.1
Rooms had views · Guests liked the comfortable beds · Guests appreciated the bathrooms

**Location** · 4.5
Shopping and sightseeing nearby · Easily accessible by car, with parking available

**Service & facilities** · 4.3
Guests enjoyed the pool · Guests spoke highly of the housekeeping, though some said the hotel management could be improved · Conference space available

# Data Cleaning

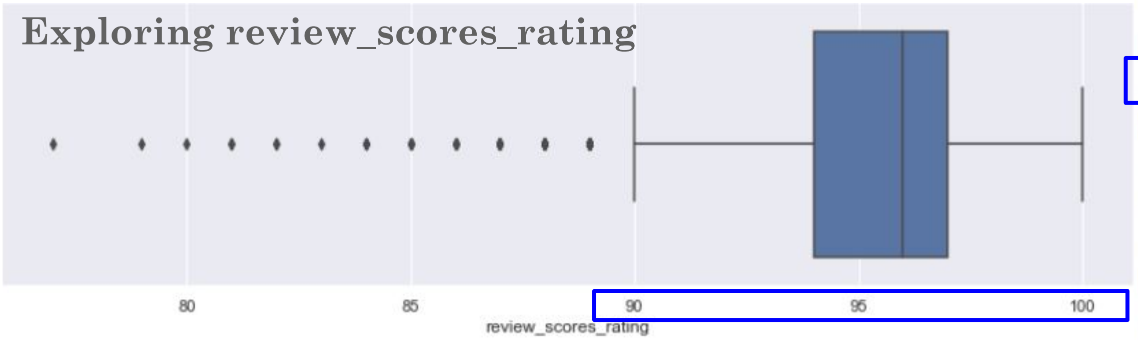
1. Removed:
   **None, NaN and** Listings with **number of review < 30**
   Unwanted chars such as “$” and “%” using Regex
   Unnecessary columns (URLs, date scraped, etc)
2. One Hot Encoding to convert categorical type to numeric type for use in the ML algorithms.
3. Obtaining amenities score (number of amenities in a listing)
4. Added summary tags for all the listing(covered later)

# Data Cleaning

Text data cleaning for sentiment analysis - lemmatizing and removing special characters from text.

```
# lower text
text = text.lower()
# removing Non-English words
text = " ".join(w for w in nltk.wordpunct_tokenize(str(text)) if w.l
# tokenize text and remove puncutation
text = [word.strip(string.punctuation) for word in text.split(" ")]
# remove words that contain numbers
text = [word for word in text if not any(c.isdigit() for c in word)]
# remove empty tokens
text = [t for t in text if len(t) > 0]
# pos tag text
pos_tags = pos_tag(text)
# lemmatize text
text = [WordNetLemmatizer().lemmatize(t[0], get_wordnet_pos(t[1])) f
# remove words with only one letter
```

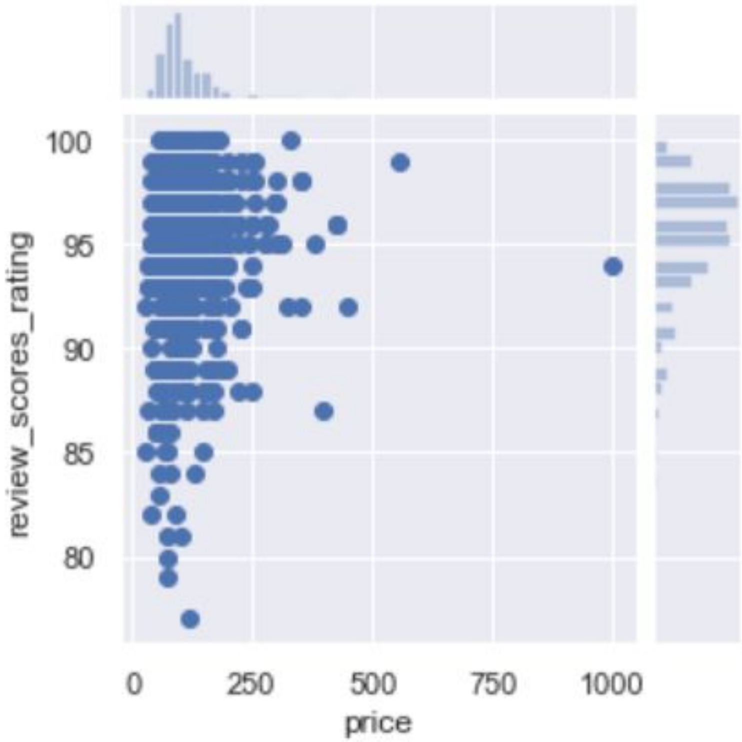
# Exploratory Analysis (Univariate)

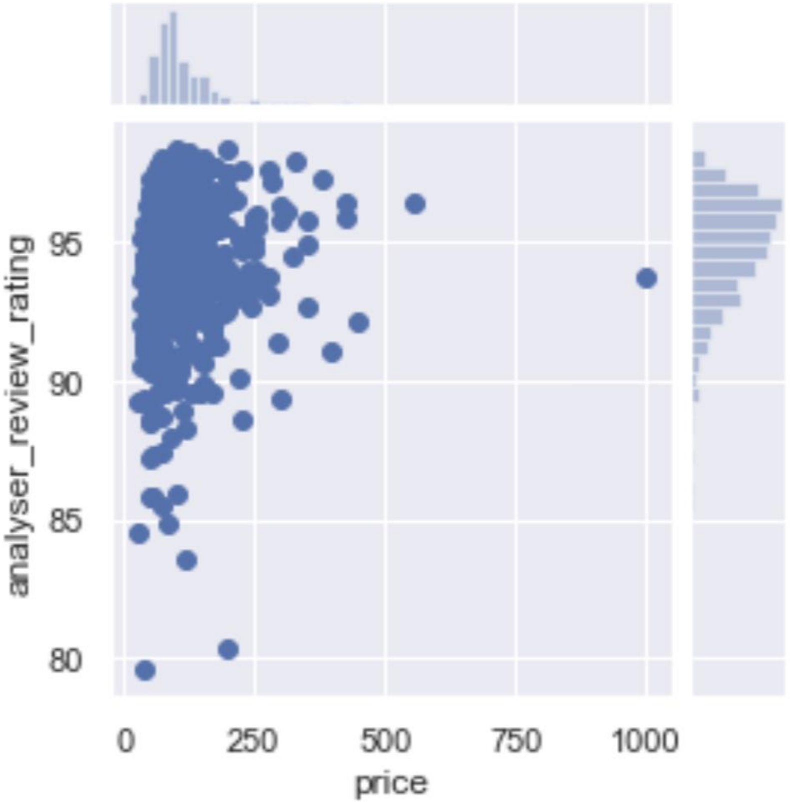



```
count     822.000000
mean       95.065693
std         3.354865
min        77.000000
25%        94.000000
50%        96.000000
75%        97.000000
max       100.000000
Name: review_scores_rating,
```

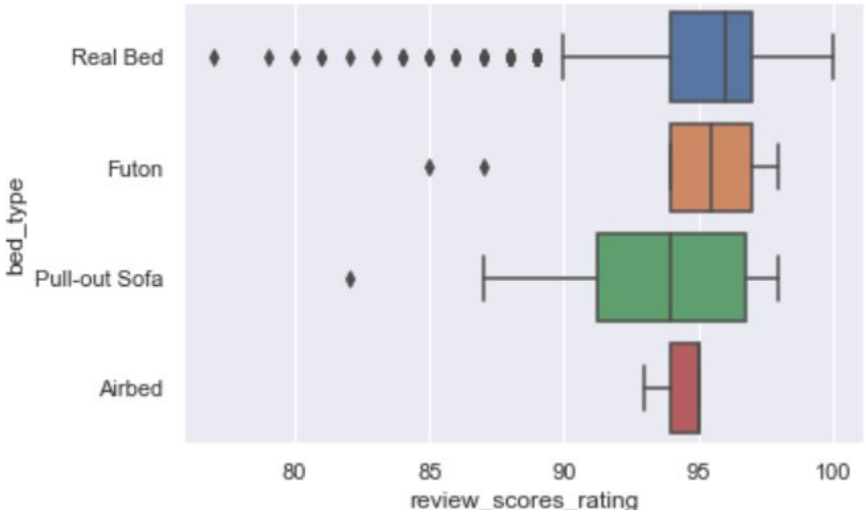

```
count     822.000000
mean       94.628467
std         2.300545
min        79.550000
25%        93.540000
50%        95.025000
75%        96.290000
max        98.350000
Name: analyser_review_rating,
```

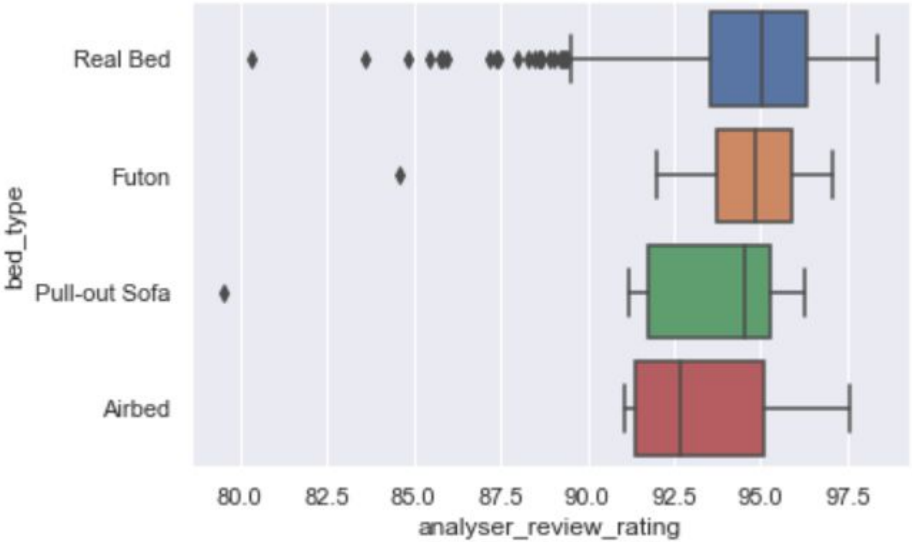
# Exploratory Analysis (Multi-variate)



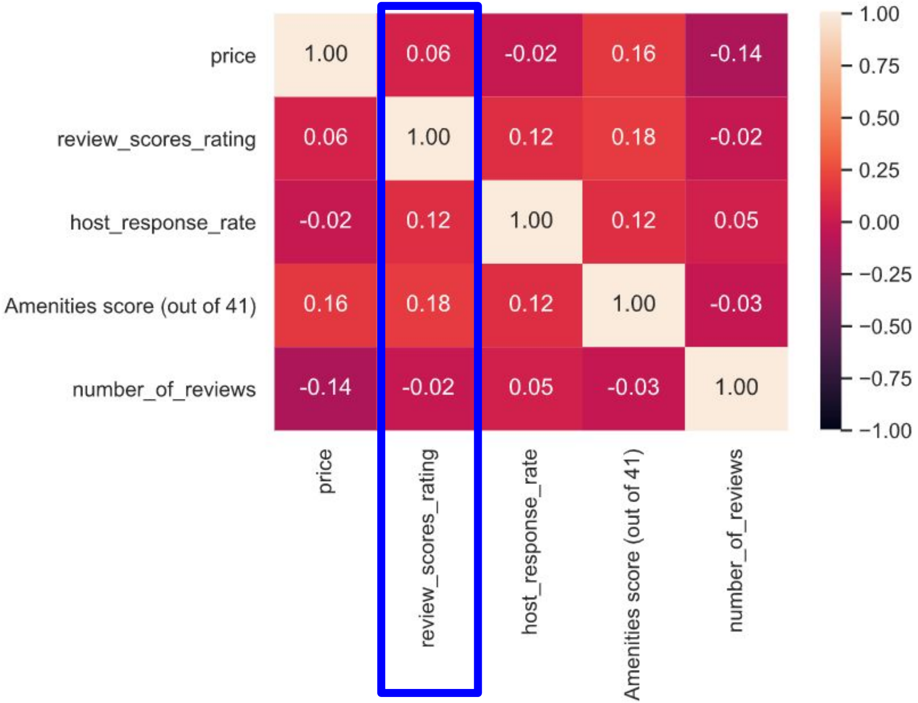



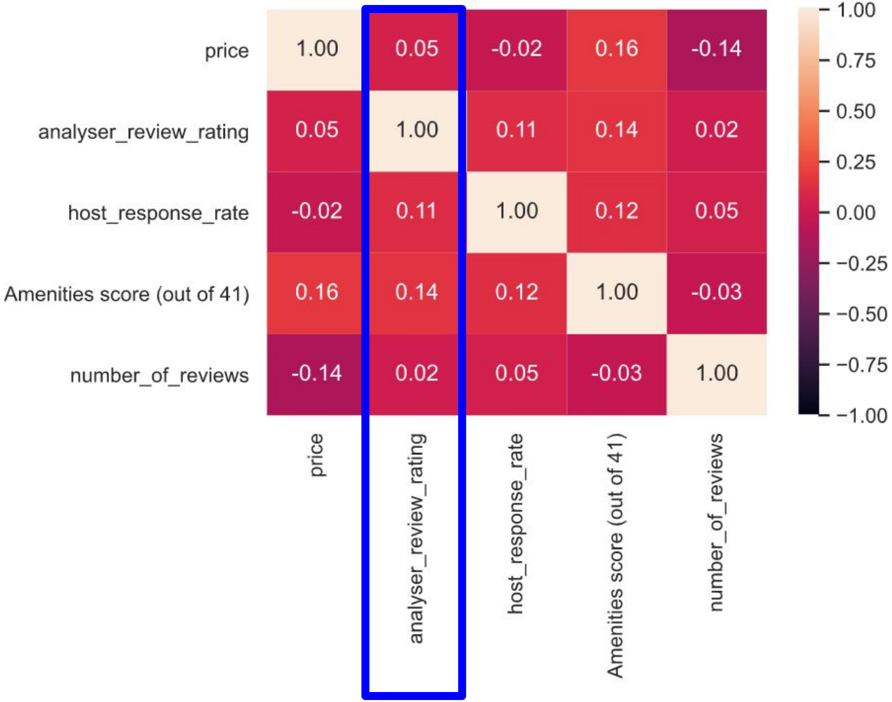
# Exploratory Analysis (Multi-variate)



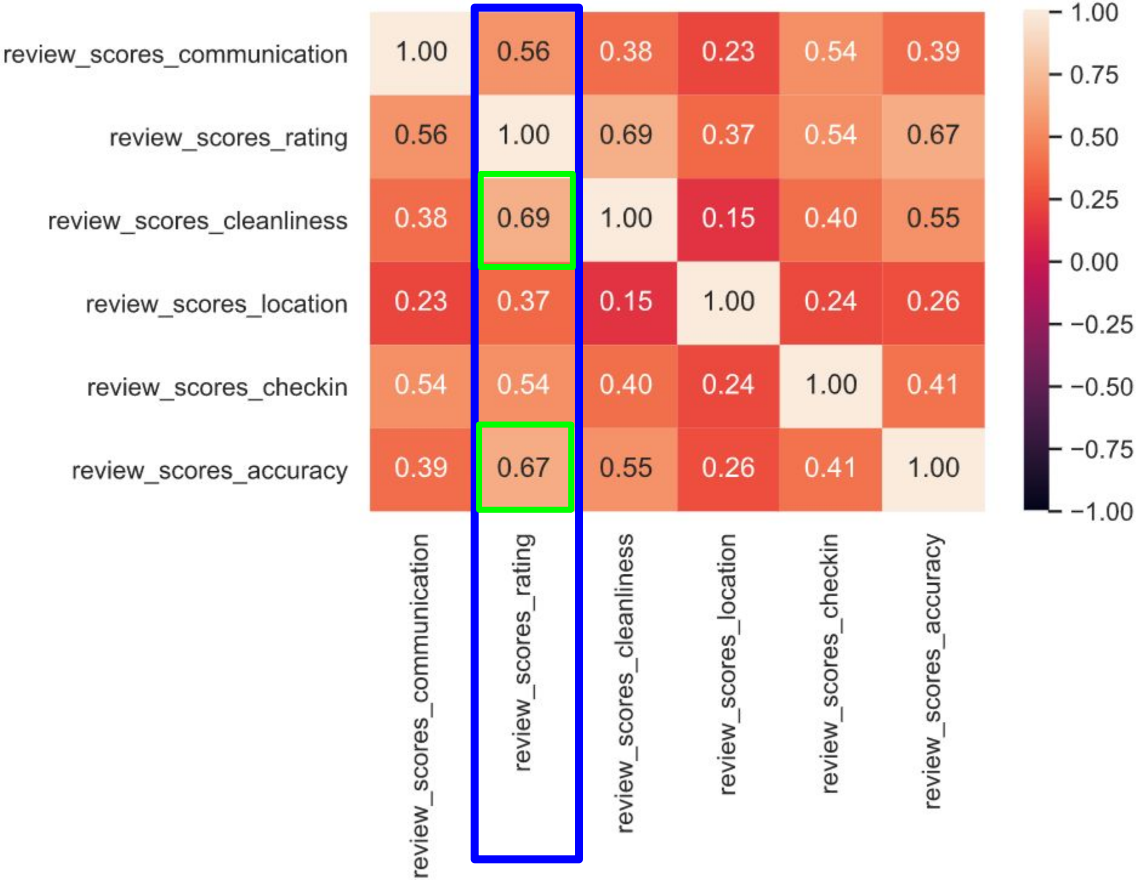



# Exploratory Analysis (Multi-variate)



# Exploratory Analysis (Multi-variate)



# Machine Learning for Model 1

The data for model 1:
Response variables: review_scores_rating, analyser_review_rating

Predictors: Listings’ features like amenities, bedrooms, bathroom, property_type etc.

Algorithms: Linear Regression, Classification

# Regression (Library: LinearRegression)

```
Goodness of Fit of Model         Train Dataset
Explained Variance (R^2)        : 0.27533913011885724
Mean Squared Error (MSE)        : 3.7392732502608 7
Mean Absolute Error (MAE)       : 1.4383207859532 09

Goodness of Fit of Model         Test Dataset
Explained Variance (R^2)        : 0.193103800369 4947
Mean Squared Error (MSE)        : 4.744483108645157
Mean Absolute Error (MAE)       : 1.5509961006490545
```

Results for analyser_review_rating

# Regression (Library: LinearRegression)

```
Goodness of Fit of Model        Train Dataset
Explained Variance (R^2)      : 0.4062999400065046
Mean Squared Error (MSE)      : 6.204485363358893
Mean Absolute Error (MAE)     : 1.7904033905054233

Goodness of Fit of Model        Test Dataset
Explained Variance (R^2)      : 0.37342886175879986
Mean Squared Error (MSE)      : 8.735696950448 5
Mean Absolute Error (MAE)     : 2.053281141639614 4
```

Results for review_scores_rating

# Classification

The response variables are numeric type so to make the classification model we made the classes using formula :

$$\text{Class} = \lfloor x / 10 \rfloor$$

Where x is the response variable and $\lfloor\ \rfloor$ represents the floor function.

Libraries used: Decision Tree Regressor and Logistic regression

# Classification

## Decision Tree:

```
Goodness of Fit of Model          Train Dataset
Classification Accuracy           : 0.9729299363057324

Goodness of Fit of Model          Test Dataset
Classification Accuracy           : 0.9235668789808917
```

## Logistic Regression:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 7 | 0.00 | 0.00 | 0.00 | 1 |
| 8 | 1.00 | 0.20 | 0.33 | 5 |
| 9 | 0.97 | 1.00 | 0.98 | 151 |
| accuracy | | | 0.97 | 157 |
| macro avg | 0.66 | 0.40 | 0.44 | 157 |
| weighted avg | 0.96 | 0.97 | 0.96 | 157 |

Results with analyser_review_rating

# Classification

Decision Tree:

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.9506369426751592

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.9235668789808917
```

Logistic Regression:

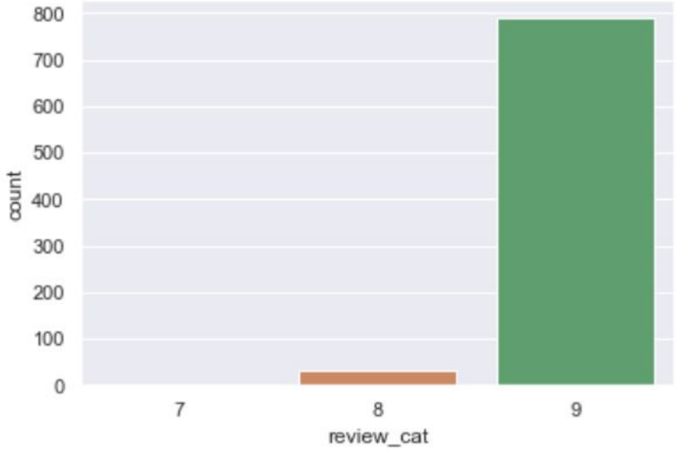
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 8 | 0.50 | 0.10 | 0.17 | 10 |
| 9 | 0.94 | 0.99 | 0.97 | 147 |
| accuracy | | | 0.94 | 157 |
| macro avg | 0.72 | 0.55 | 0.57 | 157 |
| weighted avg | 0.91 | 0.94 | 0.92 | 157 |

Results with review_scores_rating

# Classification

**But ...** the model is highly biased which is evident from f1-score and classification matrix:

```
array([[  0,    0,    1],
       [  0,    1,    4],
       [  0,    0,  151]], dtype=int64)
```




High bias in classification for analyser_review_rating

# Inferences from model 1

1. Regression model gives a good estimation in terms of mean square error and mean absolute error.
2. However, the model fails when error is compared with the variance in data.
3. Classification model yields very good results due to the highly biased nature of the data.

# Machine Learning for Model 2

The data for model 2:
Response variables: review_scores_rating, analyser_review_rating

Predictors: features based on user experience like communication, location, cleanliness etc.

Algorithms: Linear Regression, Classification, Anomaly Detection

# Regression (Library: LinearRegression)

Model 1
(previous)

| Goodness of Fit of Model | Train Dataset |
|---|---|
| Explained Variance ($R^2$) | : 0.27533913011885724 |
| Mean Squared Error (MSE) | : 3.73927325026087 |
| Mean Absolute Error (MAE) | : 1.438320785953209 |

| Goodness of Fit of Model | Test Dataset |
|---|---|
| Explained Variance ($R^2$) | : 0.1931038003694947 |
| Mean Squared Error (MSE) | : 4.744483108645157 |
| Mean Absolute Error (MAE) | : 1.5509961006490545 |

Model 2
(new)

| Goodness of Fit of Model | Train Dataset |
|---|---|
| Explained Variance ($R^2$) | : 0.35659214499502323 |
| Mean Squared Error (MSE) | : 3.1378962739072738 |
| Mean Absolute Error (MAE) | : 1.3459346209546958 |

| Goodness of Fit of Model | Test Dataset |
|---|---|
| Explained Variance ($R^2$) | : 0.41684308311685514 |
| Mean Squared Error (MSE) | : 3.780667170734291 |
| Mean Absolute Error (MAE) | : 1.3859905389060134 |

Results with analyser_review_rating

# Regression (Library: LinearRegression)

Model 1
(previous)

```
Goodness of Fit of Model           Train Dataset
Explained Variance (R^2)           : 0.4062999400065046
Mean Squared Error (MSE)           : 6.204485363358893
Mean Absolute Error (MAE)          : 1.7904033905054233

Goodness of Fit of Model           Test Dataset
Explained Variance (R^2)           : 0.37342886175879986
Mean Squared Error (MSE)           : 8.73566969504485
Mean Absolute Error (MAE)          : 2.0532811416396144
```

Model 2
(new)

```
Goodness of Fit of Model           Train Dataset
Explained Variance (R^2)           : 0.6908440185720779
Mean Squared Error (MSE)           : 3.3690519451889944
Mean Absolute Error (MAE)          : 1.4151615958257395

Goodness of Fit of Model           Test Dataset
Explained Variance (R^2)           : 0.7053250259033372
Mean Squared Error (MSE)           : 3.613316809339869
Mean Absolute Error (MAE)          : 1.553034882448464
```

Results with review_scores_rating

# Regression (Library: Random Forest)

```
R^2 train: 0.936, test: 0.696
```

Results of review_scores_rating

# Classification

Decision Tree:

Logistic Regression:

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.9665144596651446

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.9393939393939394
```

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 7 | 0.00 | 0.00 | 0.00 | 1 |
| 8 | 0.33 | 0.12 | 0.18 | 8 |
| 9 | 0.96 | 0.99 | 0.97 | 156 |
| accuracy | | | 0.95 | 165 |
| macro avg | 0.43 | 0.37 | 0.39 | 165 |
| weighted avg | 0.92 | 0.95 | 0.93 | 165 |

Results with analyser_review_rating

# Classification

Decision Tree:

```
Goodness of Fit of Model          Train Dataset
Classification Accuracy           : 0.9634703196347032

Goodness of Fit of Model          Test Dataset
Classification Accuracy           : 0.9696969696969697
```

Logistic Regression:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 8 | 0.50 | 0.25 | 0.33 | 4 |
| 9 | 0.98 | 0.99 | 0.99 | 161 |
| accuracy | | | 0.98 | 165 |
| macro avg | 0.74 | 0.62 | 0.66 | 165 |
| weighted avg | 0.97 | 0.98 | 0.97 | 165 |

Results with review_scores_rating

# Classification

Though the data is highly biased for classification, the accuracy increased by 4% for review_scores_rating, assuming the best classification library for the 2 models.

Even the f1-score showed an upward trend when compared to the results of model 1.

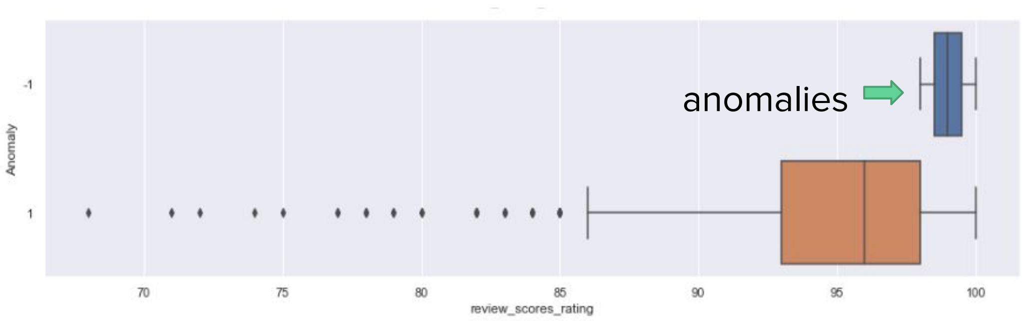
# Anomaly Detection

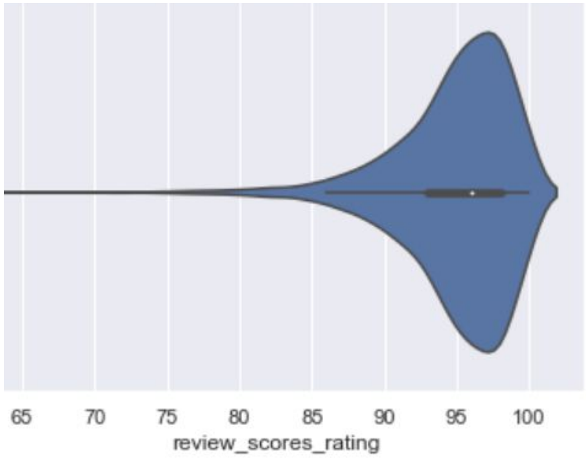
To further the analysis on model 2,

Multivariate anomaly detection was performed on the features used in model 2.

The aim was to check how the labelled anomalies would be distributed for review_scores_rating

# Anomaly Detection

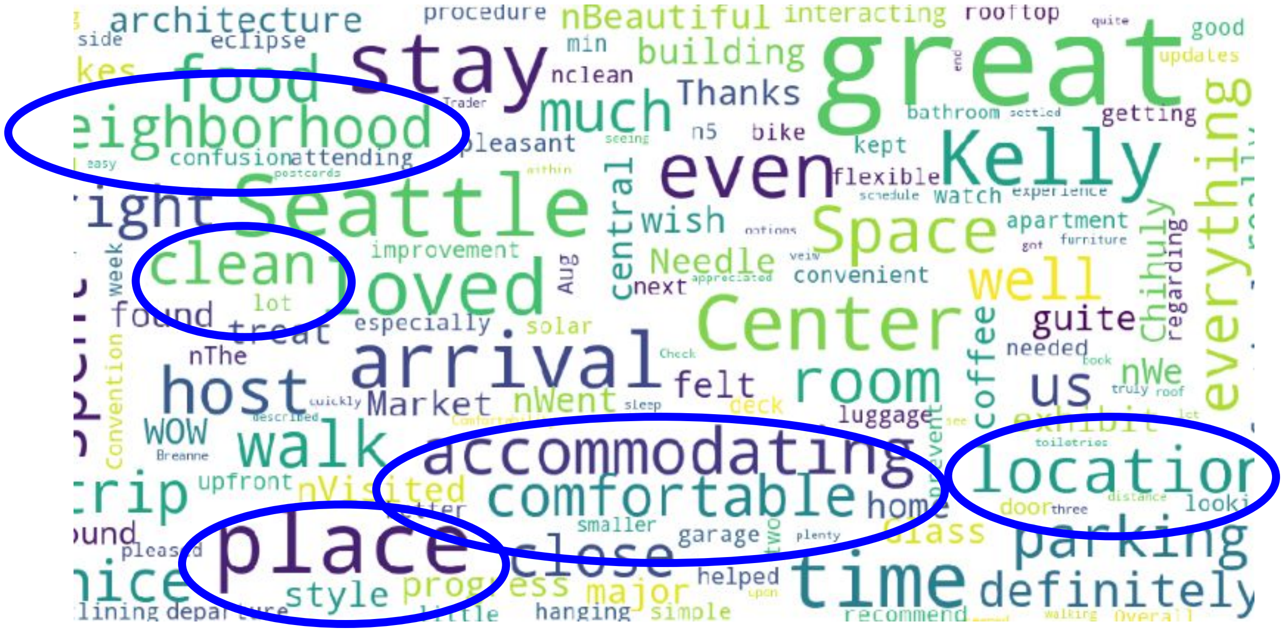


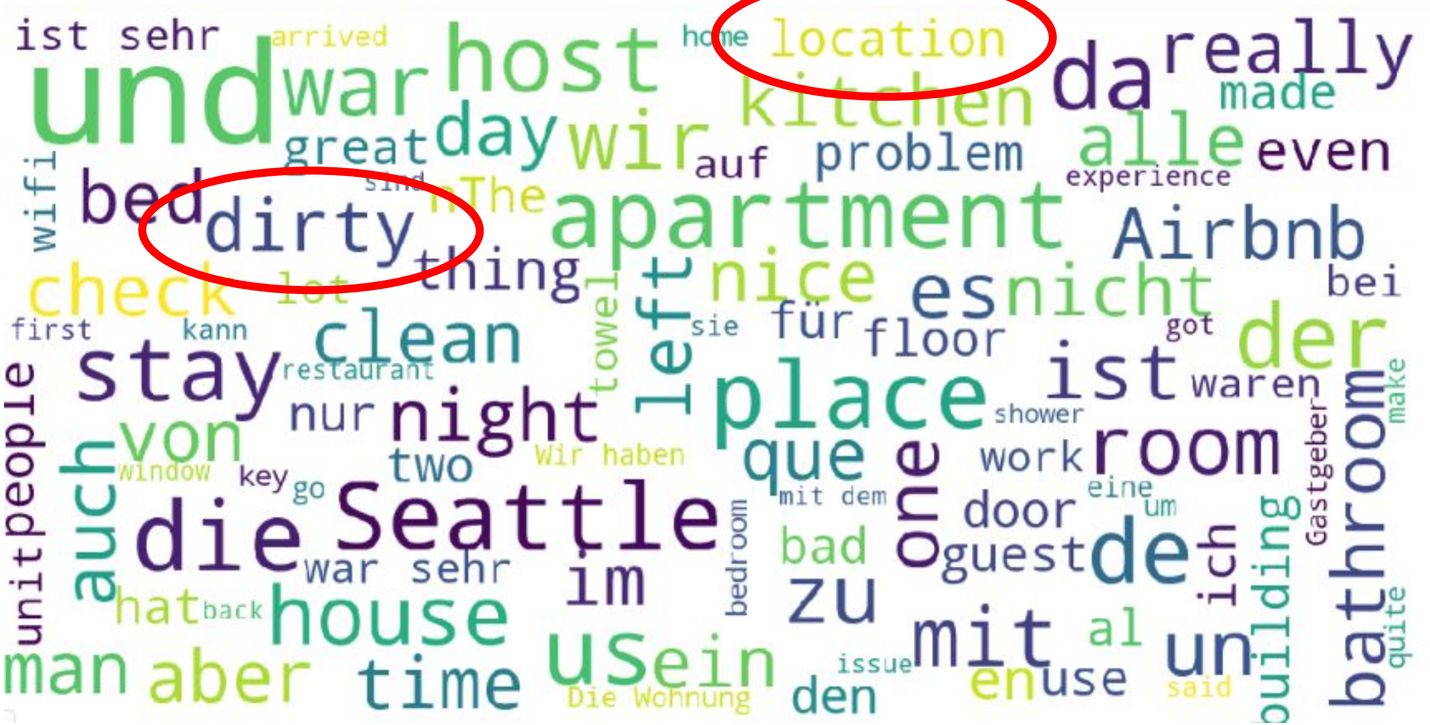

The anomalies in the features help label the top boundary points in review_scores_ratings.

# Analytic visualization

In an attempt to find a better model, we further analysed the data using Plotly and word Cloud.

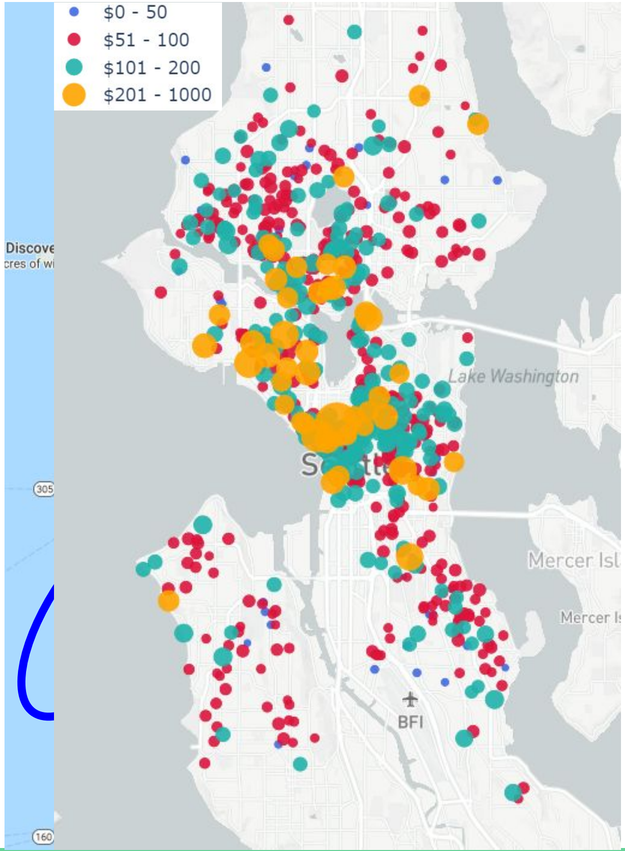


# Analytic visualization (word Cloud)

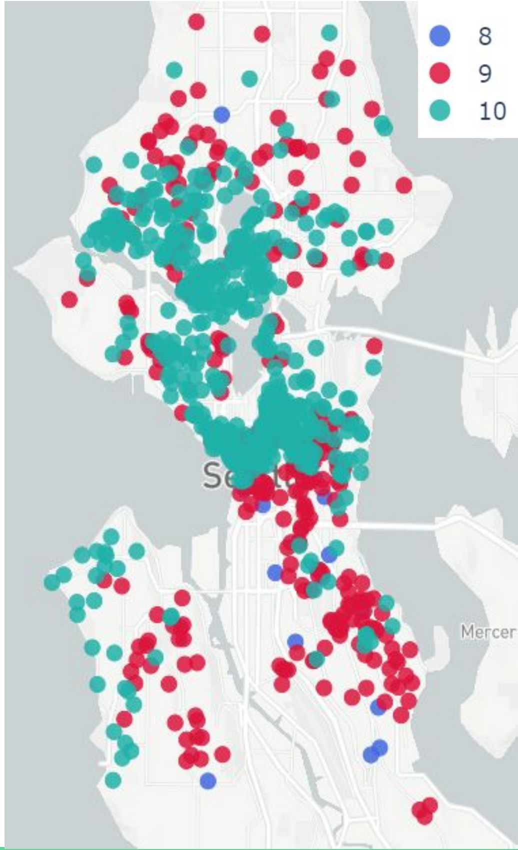


# Analytic visualization (Plotly maps)

The map visuals show the distribution of review_location on the map of Seattle.




Location rating of listings



# Conclusions

1.The second model is a better reflection of review ratings of the listings.

2. The features of the listings are not a very good reflection of the review ratings.

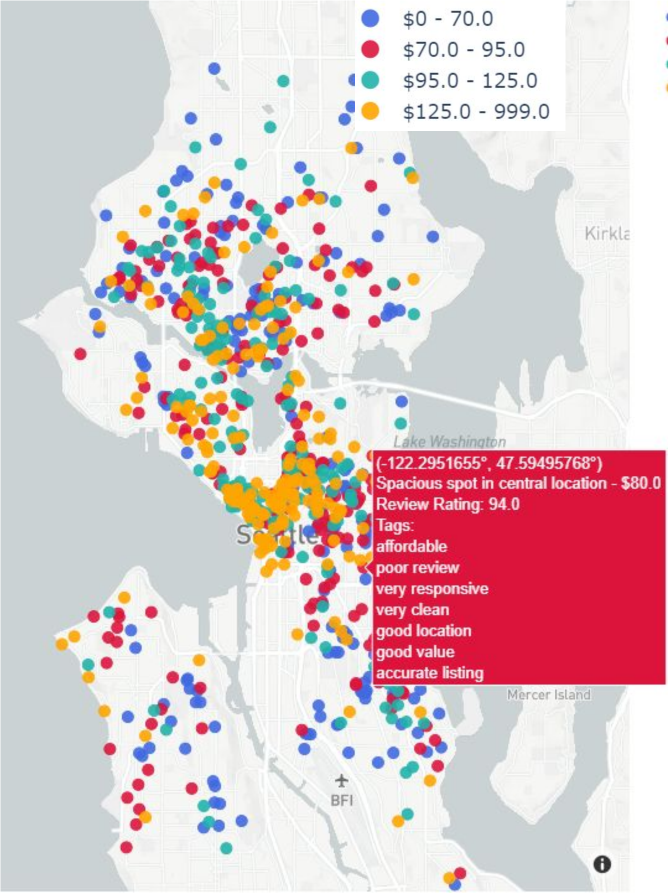
3.It was difficult to make good regression/classification models due to skewness of review score dataset.

4. Optimize review scores for hosts.

# Improving Search Experience

Added summary tags for each listing.

E.g. cleanliness is mostly rated at a **9 or 10**. Users who are not familiar with the site might believe 9 is a good rating when it is actually below average.



# Learning Points

1. Natural Language Processing and Sentiment Analysis.
2. Text Data Cleaning and Normalization.
3. Plotly and word cloud visualization.
4. Logistic regression and importance of f1-score.

# Individual Contributions

**Ashton:** Data Visualization, Regression models & Data Preparation

**Sitian:** Exploratory Analysis (Multi-variate) & Data Preparation

**Heather:** Classification Models & Exploratory Analysis(Uni-variate)

**Pratyush:** Natural Language Processing(Sentiment Analysis), Text Data Analysis & Anomaly Detection in the final model

# Thank you!