Given – the **determinants or dataset of the number of reviews each listing receives per month (reviews\_per\_month**)

**reviews\_per\_month – if missing consider 0 reviews**

Problem Statement to refute or confirm –

1. **Larger properties should receive more reviews** **because larger properties can accommodate more guests and therefore generate more traffic.**

Larger properties should receive more reviews.   
Here we need to define what is meant by larger properties because there are multiple columns like – defining a property as Larger property or smaller property or any category the board suggests.

Relationship between properties and accommodation of guests

Find correlation between accommodates and property\_type or room\_type

1. Reviews by property type
2. Max property reviews
3. Max room type reviews
4. **A property is overpriced is one of the most important factors in determining how many reviews it will receive.** In general, **the listings that are priced higher than listings of similar sizes and/or locations will receive fewer reviews than those that are priced lower.** This is because people are more likely to leave a review if they feel like **they got good value for their money**.
5. the management asks you to **build a predictive model for reviews\_per\_month**. You should **compare different models or conduct variable selections**.

Other than the factors **mentioned by the business team, what else are important predictors for reviews\_per\_month**?

Data Processing

#importing required libraries

#Import Data

# basic data exploration

#Charts n graphs

Graphs plot review vs room type

Review vs price

Roomtype and price vs review rates per month

#Correlation matrix

PDF/CDF of Review\_per\_week, MLE, Rsquare is low

Target selection training date

Feature selection

Dimensionality reduction

Model evaluation accuracy, F1, residual SSE?

1. EDA

* Box plot
* Histograms
* Distribution plot
* Aggregation for all numerical columns
* Unique values across all columns
* Duplicate values across all columns
* Correlation heatmap for all numeric variables
* Regression plot – all numeric variables
* Bar plot for all numeric variables
* Linechart to see trend of data
* Plot the skewness of all numeric variables

1. Data Cleaning

* Check for missing values in all columns
* Coinsistent formating of column data
* Remove duplicates
* Remove rows having bnegative values
* Add columns as required

1. Identifying columns related to reviews\_per\_month

* Find top 20 most reviewed location with analysis and conclusion
* Find how many are larger properties or top 50 large properties analyse with reviews per month given to each of them
* Identify trends wrt various other columns in dataset
* If required create a concentrated dataset for modeling

1. Training and testing of the dataset

Data analysis

Continuous variables

Categorical variables

Model and predictions-

Logistic Regression model – is inapplicable for this scenario since we have to predict the ratings based on the data inputs given. For Logit to be applicable the model should have the target variable y to be in range [0,1] and should have a binary classification problem to predict instead of continuous variable like the reviews\_per\_month.

**4. Performance evaluation**

* Accuracy
* Confusion Matrix
* Precision, Recall, and F1-score
* More details [here](https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c)

**6. PCA**

* **Principal Component Analysis (PCA)** is a dimension-reduction tool that can be used to reduce a large set of variables to a small set that still contains most of the information in the large set.
* The main purpose of principal component analysis is to:
  + identify hidden pattern in a data set,
  + reduce the dimensionnality of the data by removing the noise and redundancy in the data,
  + deal with multicollinearity

import numpy as np

# correlation matrix

corr = df2.iloc[:, :14].corr()

# Generate a mask for the upper triangle

mask = np.zeros\_like(corr, dtype=np.bool)

mask[np.triu\_indices\_from(mask)] = True

# Set up the matplotlib figure

f, ax = plt.subplots(figsize=(11, 9))

# Generate a custom diverging colormap

cmap = sns.diverging\_palette(220, 10, as\_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio

sns.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0,

square=True, linewidths=.5, cbar\_kws={"shrink": .5})

plt.show();

pca = PCA(n\_components = 0.8)

pca.fit(x\_train\_s)

print('Variance ratio of each pc:\n', pca.explained\_variance\_ratio\_, '\n')

print('Explained variance of each pc:\n', pca.explained\_variance\_, '\n')

print('Selected {} pcs'.format(pca.n\_components\_))

print('Original dataset shape: ', df2.shape)

|  |  |  |  |
| --- | --- | --- | --- |
| **Analysis** | Analysis is overly simplistic or incomplete | Analysis is adequate | Complete, accurate, and informative analysis |

Analysis of Data and Problem Statement—

|  |  |  |  |
| --- | --- | --- | --- |
| **Results** | Conclusions are missing, incorrect, or not based on analysis. | Conclusions relevant, but partially correct or partially complete | Relevant conclusions explicitly tied to analysis and to context |

Results based on Data analysis –

|  |  |  |  |
| --- | --- | --- | --- |
| **Organization** | The report is not eﬀectively organized. There may be parts of the report that has no explanation or no clear explanation. Graphics and tables are of poor quality | The report is overall clear but occasionally difficult to follow. Graphics and tables are included at appropriate points. | The report has a logical structure. The chain of reasoning or progression of ideas is clear. Tables and graphs are easy to follow and clearly presented |

Coding –

|  |  |  |  |
| --- | --- | --- | --- |
| **Code Readability** | Code is messy and poorly organized; unused or irrelevant code distracts when reading code. Variables and functions names do not helpful to understand code. | Code is reasonably well organized. Variable and function names generally meaningful and helpful for understanding. | Code very well organized. No irrelevant or distracting code. Variable and function names have clear relationship to their purpose in the code. Code is easy to read and understand. |
| **Code Reproducibility** | Most of the code failed to run | A few parts of the code failed to run | Code correctly loads data and generate all results and figures in the report |

|  |  |  |  |
| --- | --- | --- | --- |
| **Writing** | Explanation is illogical, incorrect, or incoherent. There are few complete sentences, only bullet points | Explanation is largely correct. Complete sentences are used. | Explanation is correct, complete, convincing, and elegant. Complete sentences are used. |

Overall maintain

This time, you'll learn to fit models that include multiple explanatory variables. This is sometimes called "multiple regression". Including more explanatory variables in the model often gives you more insight into the relationship between the explanatory variables and the response, and can provide more accurate predictions. It's an important step towards mastering regression.

**4. The course contents**

Here's the plan. In Chapter 1, you'll explore parallel slopes linear regression. This is a special case of multiple linear regression, with one numeric explanatory variable and one categorical explanatory variable. Chapter 2 introduces interactions between variables and covers Simpson's Paradox, a counter-intuitive result affecting models containing categorical explanatory variables. Chapter 3 extends linear regression to even more explanatory variables, and gives some deeper insight into how linear regression works. Finally, Chapter 4 introduces multiple logistic regression, the logistic distribution, and digs into how logistic regression works.

**5. The fish dataset**

Here's the same fish dataset from the previous course. Each row represents a fish, the mass is the response variable, and there is one numeric and one categorical explanatory variable.

**6. One explanatory variable at a time**

Recall that you run a linear regression by using ols from statsmodels dot formula dot api, passing a formula and a DataFrame. The formula has the response variable on the left and the explanatory variable on the right, with the variables separated by a tilde. You then fit the model using dot fit. Here you can see mass modeled against length. Printing the model parameters using the params attribute shows the model coefficients. With a single numeric explanatory variable, you get one intercept coefficient and one slope coefficient. Let's change the explanatory variable to species. Recall that when you have a categorical explanatory variable, the coefficients are a little easier to understand if you use "plus zero" to tell statsmodels not to include an intercept in the model. Now you get one intercept coefficient for each category. That is, one coefficient for each species of fish.

**7. Both variables at the same time**

To include both explanatory variables in the model, you combine them on the right-hand side of the formula, separated with a plus, just like you did with the zero. This time there is one slope coefficient, and an intercept coefficient for each category in the categorical variable.

**8. Comparing coefficients**

Examining the coefficients of each model, it's clear that the numbers are different. Notice that the slope coefficient for length, labeled length\_cm, changes from thirty five to forty three once you include species in the model as well. The intercept coefficients for each species show an even bigger change. For example, once you add length into the model, bream changes from six hundred and eighteen to minus six hundred and seventy two.

**9. Visualization: 1 numeric explanatory variable**

Here's the standard visualization for a linear regression with a numeric explanatory variable. Using Seaborn's regplot function, you draw a scatterplot with linear trend line, specifying the x, y, and data arguments. Setting ci to None prevents plotting a confidence interval ribbon. You print the plot using plt dot show.