

**School of InfoComm Technology**

**Machine Learning**

Diploma in Data Science (DS)

Diploma in Information Technology (IT)

October 2023 Semester

**INDIVIDUAL ASSIGNMENT 1**

(30% of Machine Learning Module)

**Deadline for Submission:**

**10th Dec 2023 (Sunday), 2359 Hours**

|  |  |  |
| --- | --- | --- |
| Student Name | : | Eng Zhen Ye |
| Student Number | : | S10243329J |
| Video Presentation Link | : |  |

**Penalty for late submission:**

10% of the marks will be deducted every day after the deadline.

**NO** submission will be accepted after 17th Dec 2023, 23:59.

Table of Contents

1. **Overview3**
2. **HR Analytics4**

2.1 Problem Understanding 4

2.2 Data Exploration 5

2.3 Data Cleansing & Transformation 9

2.3.1 Handling Null values9

2.3.2 Handling Outliers 9

2.3.3 Transforming numerical features9

2.3.4 Encoding categorical features10

2.3.5 Scaling the data11

2.4 Correlation Analysis 12

2.5 Improving the Model13

2.5.1 Dealing with outliers in ‘length\_of\_service’13

2.5.2 Removal of features 15

2.5.3 Experimentation with PCA 15

1. **Airbnb Listings 16**

3.1 Problem Understanding 16

3.2 Data Exploration 17

3.3 Data Cleansing & Transformation 22

3.3.1 Handling Null values22

3.3.2 Handling Outliers23

3.3.3 Transforming numerical features24

3.3.4 Encoding categorical features27

3.3.5 Scaling the data28

3.4 Correlation Analysis 29

3.5 Improving the Model30

3.5.1 Model Summary30

3.5.2 Trimming outliers in ‘price’ 31

3.5.3 Removing outliers in ‘price’ 31

3.5.4 Dropping of features 32

3.5.5 Implementing PCA 33

1. **Summary & Further Improvements 35**

4.1 Summary 35

4.2 Further Improvements36

1. **Overview**

In this report I will document the analysis and findings in my process of exploration and analysis through visualization and statistical approaches and then conduct data preparation to prepare the data ready for machine learning modeling.

In data exploration we will conduct a thorough analysis using statistical and visualization approaches to familiarize myself with the datasets as well as to take note of any nulls, and necessary data transformation that I need to take note of.

We will then prepare the data by cleansing and transforming the data. This includes null imputation, handling outliers, transforming categorical data, and scaling of the data. This will be done in accordance with what I have learnt about the data in the data exploration phase.

There are 2 datasets that we will be conducting in this process of data exploration and preparation.

In the first dataset, ‘hr\_data.csv’ contains employee personal information, including education background, past performance etc. I will be conducting the aforementioned data exploration and preparation to prepare the data for machine learning modelling. The goal of this is to predict whether an employee is likely to get promoted or not, therefore being a classification problem.

The second dataset, ‘listings.csv’ consists of host information, the condition of listed properties, the reviews etc. Exploring and then preparing the data for machine learning modelling with the goal of predicting the rental price of the listed properties. Hence, this is a regression problem.

At the end of the report, I will summarize our findings and explain possible further improvements that could have been made.

1. **HR Analytics**
   1. **Problem Understanding**

Human Resources (HR) departments are leveraging data analytics to enhance the

way they operate, resulting in higher efficiency and better results overall. Traditionally, HR analytics and processing has been manual which, due to the nature of human resources dynamics, constrained the scope of insights and outcomes.

However, in this scenario we will be making use of machine learning to predict employee promotions based on various HR-related factors.

The dataset ‘hr\_data.csv’ comprises of the following variables:

|  |  |
| --- | --- |
| **Variable** | **Definition** |
| employee\_id | Unique ID for employee |
| department | Department of employee |
| region | Region of employment (unordered) |
| education | Education Level |
| gender | Gender of Employee |
| recruitment\_channel | Channel of recruitment for employee |
| no\_of\_trainings | no of other trainings completed in previous year on soft skills, technical skills etc. |
| age | Age of Employee |
| previous\_year\_rating | Employee Rating for the previous year |
| length\_of\_service | Length of service in years |
| KPIs\_met >80% | if Percent of KPIs(Key performance Indicators) >80% then 1 else 0 |
| awards\_won? | if awards won during previous year then 1 else 0 |
| avg\_training\_score | Average score in current training evaluations |
| is\_promoted | (Target) Recommended for promotion |

**Problem Statement:** The objective is to build a predictive model that can accurately classify employees as ‘promoted’ or ‘not promoted’ based on the provided features. The model will assist HR departments in identifying employees with high potential to be promoted, enabling focused HR strategies and talent management decisions.

* 1. **Data Exploration**

I first plotted a bar chart to explore the proportion of promoted and not promoted. My

A bar chart with blue squares

Description automatically generatedrationale for plotting this bar chart the first thing is to look if there is an imbalance in our target variable. Imbalance in the target variable for the machine learning model can introduce bias and skew our model accuracy.

From the bar-plot above it is easy to infer that there is a clear imbalance in the number of 0s and 1s. Upon seeing this I quickly did stratified sampling, a sampling technique to balance counts based on one or more variables. In this case it reduced the count of 0s to balance it with the number of 1s. Below shows the count of 1s and 0s in ‘is\_promoted’ in the end.



Next, I explored the total count of rows of each feature as well as the data type of each feature. This is done so that I can ensure that each data type for each feature is correct.

A screenshot of a computer

Description automatically generated

Additionally, I counted the number of nulls in each feature so as to know which features have nulls as well as the proportion of nulls in each feature.

A screenshot of a computer screen

Description automatically generated

From the image above it is clear that there are null values in ‘education’ and ‘previous\_year\_rating’. In ‘education’ it is entirely possible that it is null due to the employee not having an education. In ‘previous\_year\_rating’ there is a sizable number of nulls. However, this is most likely due to the employee being new to the company and therefore having no rating in the previous year.

A graph of a box plot

Description automatically generated After seeing the results above, I went to explore the features that had null values.

A graph with blue squares

Description automatically generated

In ‘education’ there is a significant number of employees having a Bachelor’s and Master’s & above but not a lot of employees with Below Secondary. Knowing that the count of nulls in ‘education’ were sizably smaller than the other categories in the feature it probably would not affect the machine learning model significantly to add a new category called ‘no education’ and might even be detrimental to the model.

In ‘previous\_year\_rating’ I noticed that the range was from, not 1 to 5, but 0 to 5. This gave me the idea of imputing the nulls with 0 in this feature, that way it is separate from the other values.

A diagram of a graph

Description automatically generatedAfter this I then plotted box plots for every numerical value to check for glaring outliers. Outliers are important to handle in machine learning models as it can introduce noise, skew coefficients, make the model bias, distort relationships and overall impact accuracy and reliable of the machine learning model. Below are some boxplots or features that I found to be intriguing.

A diagram of a line graph

Description automatically generated

I notice that there are outliers in the ‘length\_of\_service’ and ‘age’. ‘length\_of\_service’ having a large range of outliers. This skews the distribution of the variable and could cause it to deviate from the expected distribution which could impact the performance of the model. However, in this case, the model that we are using, logistic regression is more robust to outliers due to several reasons that I would not go through here. This means that we could handle most of the outliers in ‘length\_of\_service’ or all of them.

The feature ‘age’ only has outliers in the age range of the elderly and does not have as significant of a range of outliers. Handling outliers in this feature could be as simple as just capping the outliers.

I wrote a function that helps to plot a histogram, a Q-Q plot (quantile-quantile plot) and boxplot. This combination helps to analyze the different aspects of data distribution such as the shape, central tendency, presence of outliers and deviation from normality.

Q-Q plots checks if data points follow a perfect bell-shaped curve (similar to normal distribution) If the points follow along the line it matches a normal distribution and if the points are far from the line the data point does not follow a normal distribution.

The goal of plotting a Q-Q plot here is to see if the features follow a normal distribution and if it does not, we might use transformation techniques to make it follow a normal distribution. Normalizing features in machine learning can provide different benefits for the model such as improving model performance by bringing all features to similar scale, faster convergence during optimization, assisting in regularization preventing overfitting, facilitating interpretability by keeping coefficients consistent, and improve model robustness by preventing outliers from having too much of an effect.

Using this I plotted the numerical features ‘age’, ‘length\_of\_service’ and ‘avg\_training\_score’.

A graph with a line drawn on it

Description automatically generated The above shows the diagnostic plot of ‘length\_of\_service’, it is severely right-skewed, as expected from the boxplot done before hand. The Q-Q plot shows that it severely deviates from a normal distribution. I may use a transformation to make it follow a normal distribution instead of forcibly removing the outliers through other methods.

Lastly for data exploration I used the chi-square test to determine the significance of ‘recruitment\_channel’. The chi-square test is a statistical test used to determine whether there is a significant association between categorical variables. The test will output two results, a p-value and a chi-square statistic. The lower the p-value (typically lower than 0.05) the more it suggests that there is a significant association between the variables and the higher the chi-square statistic indicates that there is a stronger association between the variables. Below are my results.

A graph of a bar chart

Description automatically generated with medium confidence

Notice that the two variables have a chi-square value of 8.56 (3s.f.) and a p-value of 0.0138 (3s.f.). The p-value of 0.0138 is less than a typical significance level of 0.05, suggesting that the observed relationship or a more extreme relationship between the categorical variables would occur by random only 1.38% of the time if there were no association between the variables. The chi-square value of 8.56 indicates that there is a moderate discrepancy between the observed and expected frequencies. Therefore, concluding that there is a significant association between ‘is\_promoted’ and ‘recruitment\_channel’.

* 1. **Data Cleansing and Transformation**

I first dropped ‘employee\_id’ as I could not identify any sort of relationship with the other featuresand could just be noise to the model.

**2.3.1 Handling null values**

During the data exploration stage two features containing nulls were identified and explored, it being ‘education’ and ‘previous\_year\_rating’

For the feature, ‘education’, I noticed that the number of nulls were insignificant and could be resolved by imputing the mode (most frequent category). Using this method, the nulls were imputed with “Bachelor’s”

For ‘previous\_year\_rating’, in the context of HR I identified the possibility that the employees could be newly joined and therefore have no rating from the previous year and also know that the range of values in ‘previous\_year\_rating’ ranges from 1 to 5. Therefore, I decided to impute the nulls with the value of 0.

**2.3.2 Handling outliers**

From the data exploration stage we also identified and explored two features containing outliers which are ‘age’ and ‘length\_of\_service’. However, in this section we only be resolving outliers in ‘age’ and not ‘length\_of\_service’ as there will be an attempt to resolve the outliers in ‘length\_of\_service’ using transformation techniques.

To handle outliers in ‘age’ I used winsorization with the capping method ‘quantiles’, a tail of right to apply it to the right side of the distribution where the outliers are and a fold of 0.039 as I found that this was the most ideal fold to resolving outliers in this feature.

Below is the diagnostic plot of ‘age’ after the winsorization.

A graph with a red line

Description automatically generated

**2.3.3 Transforming numerical features**

The purpose of transforming is to ensure that numerical features follow a normal distribution as much as possible for a multitude of beneficial reasons that I stated in the data exploration stage.

There are 3 features in the dataset that are numerical and continuous (except for ‘employee\_id’), them being, ‘length\_of\_service’, ‘age’, and ‘avg\_training\_score’.

I experimented with various transformation techniques such as log transformation, BoxCox transformation and others to try and figure out which is the best transformation for the features to attain a normal distribution. In the end, I settled for no transformation for ‘avg\_training\_score’ since majority of values already followed a normal distribution and transformation techniques did not have an effect on it and power transformation using square root for ‘age’ and ‘length\_of\_service’.

Due to word count limit, I will not go through all the different transformations but I will explain what power transformation (square root) does. Power transformation, specifically using square root involves taking the square root of variable values. It is often used to reduce the impact of extreme values and can make highly skewed distributions more symmetrical. Which is why I decided not to handle outliers in ‘length\_of\_service’ earlier on. Below shows the before and after of ‘length\_of\_service’.

A graph with a line and a red line

Description automatically generatedBefore:

A graph with a red line

Description automatically generatedAfter:

**2.3.4 Encoding categorical features**

Machine learning models typically are not able to take in variables that are not numeric, therefore we have to encode the string values in the dataset into numerical values.

There are 5 categorical variables that we will have to encode, them being ‘department’, ‘region’, ‘gender’, ‘education’, and ‘recruitment\_channel’.

I used frequency encoding to encode ‘department’ and ‘region’ as they had potentially valuable information that could be lost by encoding especially when there were quite a few categories in both features. By using frequency encoding, it helps to preserve categorical information by replacing the categories with their count of occurrence.

I used label encoding to encode ‘gender’ as there were only two categories.

I then used ordinal encoding to encode 'education’. I felt that this was appropriate because there is a hierarchy since a “Bachelor’s” is the highest education followed by “Master’s & above” and lastly, ‘Below Secondary’. To preserve the hierarchy

Lastly, I used target encoding to encode ‘recruitment\_channel’. From the data exploration stage, I identified that ‘recruitment\_channel’ has significance in relationship with the target ‘is\_promoted’. Therefore, I decided that perhaps using target encoding will be better to capture this relationship. Target encoding, which maps categories to the mean of the target variable, is especially useful when there is a discernible association between categorical variables and the target. This enables the model to capture the nuanced variations between ‘recruitment\_channel’ and could potentially enhance the predictive performance by leveraging the inherent relationship.

**2.3.5 Scaling the data**

There are many different scalers that help to scale the data but, in most cases, a standard scaler is used due to its effectiveness. It scales the data to have a mean of 0 and a standard deviation of 1, centering the data around 0 and making it unit variance, preserving the statistical properties of the data.

Below is the before and after effect of the values.

A comparison of a line graph

Description automatically generated with medium confidence

I then ran a logistic regression model with 10,000 iterations to get a baseline train and test accuracy. In the end I got a train accuracy of 0.746 and test accuracy of 0.721. This indicates that the model is overfitting.

**2.4 Correlation Analysis**

In this section I plotted the “Pearson Correlation of Features” and a scatter matrix.

The “Pearson Correlation of Features" show that there is a lot of features in relation

to ‘is\_promoted’ have correlation values close to 0. This suggests that there is a weak linear relationship between all the features and the target. This implies that there is a low correlation and may not be ideal for linear modeling.

A screenshot of a computer

Description automatically generated

From the scatter matrix, it is hard to tell if there is a relationship between ‘is\_promoted’ and the other numerical features. However, it is quite clear that employees that have high scores in ‘avg\_training\_score’ are very likely to get promoted.

A screenshot of a graph

Description automatically generated

**2.5 Improving the Model**

In this section I will be making changes to the dataset in an attempt increase

model performance.

**2.5.1 Dealing with outliers in ‘length\_of\_service’**

First, I got the model summary, looking at the P>|z| value:

A table of numbers and symbols

Description automatically generated

Differing P-values can mean different things for the individual features.

•P>|z| ≥ 0.05: This is generally considered as evidence that the null hypothesis cannot be rejected. It suggests that the observed data is not statistically significant and provides weak evidence against the null hypothesis.

•0.01 ≤ P>|z| < 0.05: This is considered borderline significant. It suggests that there might be some evidence against the null hypothesis, but it is not conclusive. Further investigation may be warranted.

•P>|z| < 0.01: This is considered statistically significant. It suggests that the observed data is highly unlikely to occur by chance if the null hypothesis were true. This provides strong evidence against the null hypothesis and supports the alternative hypothesis.

Before making any significant changes to the dataset however I first experimented

with dealing with the outliers in ‘length\_of\_service’ as, previously, even after transformation it did not fully deal with the outliers.

A graph with a red line

Description automatically generated After using winsorization this was the result:

Running the model after this, the train accuracy increased but the test accuracy decreased with an accuracy of 0.74767 (5s.f.) and 0.72260 (5.s.f.) respectively, these changes increased overfitting but not by a lot.

I then ran the model summary. According to the P-value, ‘gender’, ‘recruitment\_channel’ and ‘length\_of\_service’ are not statistically significant to predicting the target.

A table of numbers and symbols

Description automatically generated

**2.5.2 Removal of features**

So, this time I decided to drop the three features. Running the model this time yielded me a train accuracy of 0.74751 (3s.f.) and test accuracy of 0.72189 (5s.f.). This latest model’s accuracy is slightly worse than that of the previous model.

Seeing the relatively low p-value of ‘length\_of\_service’, I decided to experiment

leaving it in the train data. This instead yielded me a train accuracy of 0.74797 (5s.f.) and a test accuracy of 0.72296 (5s.f.) which is the best model accuracy yet.

**2.5.3 Experimentation with PCA**

Lastly, I decided to experiment with PCA on the train dataset. PCA stands for Principal Component Analysis, which is commonly a dimensionality reduction technique more commonly used in data analysis and machine learning to simplify the complexity of high-dimensional datasets while retaining important information. It is due to this very characteristic that I decided to experiment with this.

PCA identifies the principal components in which the data varies the most. These components are ordered by the amount of variance they explain in the dataset. This allows for lesser dimensions in the data while still retaining the essence of the original data.

We can see these components and their variance in the following line graph:

A graph with a line

Description automatically generated

We can also see how many components there are and their respective variance in

text form. From the below image, we can see that there are 12 principal components and their respective variance.

****

Running this into the model gives me a train accuracy of 0.74644 (5s.f.) and a test

accuracy of 0.72117 (5.s.f.). This model performance is slightly worse than that of our first baseline model.

With this I concluded the HR analytics modelling and exported the data of the model

without ‘gender’ and ‘recruitment\_channel’ and with ‘length\_of\_service’ remaining in the dataset. It is saved as ‘hr\_export.csv’.

1. **Airbnb Listings**
   1. **Problem Understanding**

The goal of this machine learning model is to leverage the dataset to make

predictions on the rental price of the listed properties. The model could potentially serve to help both guests and hosts by providing insights into factors influencing property rental prices, to help them make informed decisions in Airbnb rentals.

Therefore, in this scenario we will be making use of machine learning to predict Airbnb prices.

The dataset ‘listings.csv’ comprises of the following variables:

|  |  |
| --- | --- |
| **Variable** | **Definition** |
| id | listing ID |
| name | name of the listing |
| host\_id | host ID |
| host\_name | name of the host |
| neighbourhood\_group | region |
| neighbourhood | sub region |
| latitude | latitude coordinates |
| longitude | longitude coordinates |
| room\_type | listing space type |
| price | (Target) daily rental price in dollars |
| minimum\_nights | amount of nights minimum |
| number\_of\_reviews | number of reviews |
| last\_review | latest review |
| reviews\_per\_month | number of reviews per month |
| calculated\_host\_listings\_count | amount of listing per host |
| availability\_365 | number of days when listing is available for booking |

**Problem Statement:** The objective is to build a predictive model that can accurately

predict prices of Airbnb listings with a good performance. The model will help users to make informed decisions on how to price different listings.

* 1. **Data Exploration**

I first wanted to find out if I wanted to utilize all the samples based on the data

categories so that I could make the prediction model more specialized on different categories.

To do this I loaded the data some boxplots. I plotted box plots of price vs

‘neighbourhood’, ‘neighbourhood\_group’, ‘room\_type’.

A graph of a number of black dots

Description automatically generated with medium confidenceA graph of different colored lines

Description automatically generated,

A graph of a number of numbers

Description automatically generated with medium confidence

Looking at the distributions I wanted to filter the dataset by ‘Central Region’ in

‘neighbourhood\_group’ since it seems to have the biggest range (not including the outliers) Therefore, I plotted a frequency bar plot of neighbourhood\_group.

A graph with blue bars

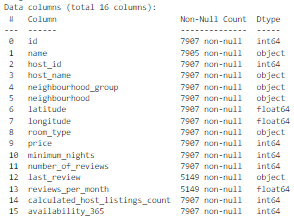
Description automatically generated

Looking at ‘Central Region’, it had the most number of rows in the data. Therefore, I

decided to filter the data to only contain ‘Central Region’. However, during the improvement of the model stage, I found that this change had a severely negative impact on the model’s performance and did not go through with this change. I suspect that this is due to the lack of data that the dataset has even with 6 thousand rows in ‘Central Region’

I suspect that for this problem, due to it being a regression problem, even with 6 thousand rows of data, the model is unable to learn the data. Therefore, I decided to revert this change.

Next, I explored the total count of rows of each feature as well as the data type of each feature. This is done so that I can ensure that each data type for each feature is correct.



Additionally, I counted the number of nulls in each feature so as to know which

features have nulls as well as the proportion of nulls in each feature.

A screenshot of a computer program

Description automatically generated

From the image above it is clear that there are null values in ‘last\_review’,

‘reviews\_per\_month’ and ‘name’. In ‘last\_review’ and ‘reviews\_per\_month’ I considered the possibility that it is null due to the listing just simply not have any reviews.

I then went to explore the features that had nulls.

I first looked at the different categories that ‘name’ had but I found out that they

seemed to be just descriptions of the listings. I then considered if it could be used for feature engineering, but the descriptions were extremely inconsistent as some had the word “BR” standing for bedroom while others do not, and some had “… min walk from MRT” while others do not. Due to these inconsistencies, I decided that this feature was not worth feature engineering therefore I dropped the feature.

Next, I looked at a box plot of ‘reviews\_per\_month’.

A diagram of a box plot

Description automatically generated

A screenshot of a computer

Description automatically generated

The distribution shows a wide range of outliers ranging from 3 to 13 with seemingly a

lot of outliers as well, skewing the distribution drastically. However, further exploration using descriptive statistics I found out that the minimum value is 0.01 and not 0. Since there is a possibility that there are simply no reviews and therefore null values, I have decided to impute the nulls later with 0 later on.

I was unable to explore ‘last\_review’ as it is a datetime value, however since there is a possibility that there is simply no last review due to the listing not having any reviews, I decided that when I split this feature into day, month, and year. I will simply impute these with 0.

Next is plotting the outliers in the numerical features of the dataset, since there are

many numerical values, I will display the ones that have outliers or are notable:

A diagram of a graph

Description automatically generatedA diagram of a diagram

Description automatically generated

A diagram of a graph

Description automatically generated with medium confidenceA diagram of a number of reviews

Description automatically generated

The above show the features ‘latitude’, ‘longitude’, ‘minimum\_nights’ and

‘number\_of\_reviews’ and they are features that contain a large amount of outliers. In addition to the above there is also ‘reviews\_per\_month’ which was the feature that we viewed when exploring features with nulls.

Handling these outliers can heavily affect the linear regression’s performance negatively. In linear regression, the model attempts to find the best-fit line through the data. Due to outliers being far from the majority of data points it can disproportionately affect the model’s coefficients (slope and intercept), affecting the accuracy when predicting new observations. This is especially so when the extreme values might not be representative of the general pattern in the data.

However, the following is somewhat more concerning where the large range of

outliers are present in ‘price’, our target variable.

A diagram of a price

Description automatically generated

The above is a box plot of ‘price’ and it contains a very wide range of outliers. The

extreme prices can excessively pull the model’s predictions, causing biased estimation especially when many data points are concentrated in the lower prices. Moreover, the outliers also violate the assumptions of the linear regression model.

Since we should not be transforming the target, I am only left with forcibly removing the outliers instead to preserve model accuracy. However, this could also remove valuable information.

Next, I plotted the diagnostic plots of the numerical features to investigate whether

they follow a normal distribution. Linear regression models assume that the numerical values have a linear relationship with the target variable and that the residuals are normally distributed. However, the normality of the data is not a strict assumption, but it can still impact the interpretability of coefficients and can influence the overall model performance.

Numerical features that follow a normal distribution still allows a more straightforward interpretation of coefficients for the model and can better represent the relationships between the data and the target.

There are many numerical variables that do not follow a normal distribution as well. In fact, almost all features except for ‘id’ and ‘host\_id’ do not follow a normal distribution closely. The features that glaringly do not follow a normal distribution as closely as I would like are ‘minimum\_nights’, ‘number\_of\_reviews’ and ‘reviews\_per\_month’.

‘minimum\_nights’:

A graph with a line and a line

Description automatically generated with medium confidence

‘number\_of\_reviews’:

A graph with a line

Description automatically generated

‘reviews\_per\_month’:

A graph with a blue line

Description automatically generated

I suspect that many of the outliers in the numerical features that contribute to the

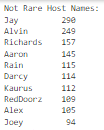
data not being normal also come from the outliers in ‘price’ as well. However, I am willing to experiment with the linear regression model to see if the model is able to learn the relationship between the data and ‘prices’ before removing them forcibly.

Lastly, I noticed that ‘host\_name’ contains many different categories. There were so

many that I decided to explore the unrare categories of the dataset and how much of the dataset they constitute. The below image shows the most common host names with the threshold for rare categories as 1%. If the threshold is any higher, there will be a lot less host names on display

A graph with numbers and names

Description automatically generated with medium confidence



The graph also depicts the distribution of count of host names within the feature, with

only a small percentage of host names being a non-rare category. With the many host names, I am willing to experiment with rare encoding ‘host\_name’, however I still do not think that this will be a significant feature in predicting ‘price’.

* 1. **Data Cleansing and Transformation**

As a note, I dropped the ‘name’ column during the data exploration phase as I was

unable to find any significant data that I could feature engineer from it and therefore rendered it not useful.

**3.3.1 Handling Null Values**

From the data exploration section there were null values from ‘last\_review’ and

‘reviews\_per\_month’ and that we have identified that the null values in both features could be due to the simple fact that the listings did not have reviews. With this information I decided that I would impute the nulls with the number 0.

First, I converted ‘last\_review’ to the datetime data type and then split it into different features of ‘day’, ‘month’ and ‘year’. Since ‘day’, ‘month’ and ‘year’ had null values still, I then imputed them with 0s.

I considered doing feature engineering in ‘last\_review’ but thought that it might introduce noise into the data and did not do it. I then dropped the column ‘last\_review’.

Next, I imputed ‘reviews\_per\_month’ with 0.

The following is the result of handling the null values and creating ‘day’, ‘month’ and ‘year’.

A screenshot of a computer program

Description automatically generated

**3.3.2 Handling outliers**

For the handling of outliers, I first trimmed the far out and large values of ‘price’ by

dropping rows with a ‘price’ of more than 8000. This was done so that I could then plot out the other features using diagnostic plot without these big outlier values.

Which I then plotted the diagnostic plot of every numerical feature in my dataset to look for outliers. The features containing outliers are ‘latitude’, ‘longitude’, ‘price’, ‘minimum\_nights’, ‘number\_of\_reviews’, ‘reviews\_per\_month’ and ‘calculated\_host\_listings\_count’. Out of these, however, we will only be winsorizing ‘latitude’, ‘longitude’, ‘minimum\_nights’ and ‘calculated\_host\_listings\_count’ in hopes that the other features would be better handled using the transformers in the next section. I will then insert them into model if the model can detect any relationship between the features and the target and experiment with forcibly handling the outliers.

For all the winsorizers, I used the capping method ‘quantiles’ and a fold of 0.05,

0.05, 0.01 and 0.07 for ‘latitude’, ‘longitude’, ‘minimum\_nights’ and ‘calculated\_host\_listings\_count’ respectively as I found that they were the best configuration to trim the outliers that was desirable to me. The following images are the diagnostic plots of the 4 features handled here.

A graph with a line and a red line

Description automatically generated

A graph with a line going up

Description automatically generated

A graph with a line

Description automatically generated

A graph with a red line

Description automatically generated

1/

**3.3.3 Transforming Numerical Features**

In this section, I will be performing transformations on numerical features that do not follow a normal distribution.

I once again experimented with the different transformation techniques on the several numerical features including reciprocal transformation and Yeo Johnson Transformation.

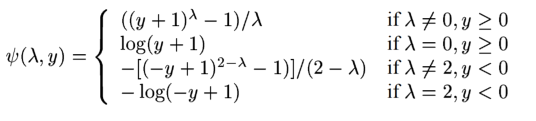
After experimentation, I settled with ‘latitude’ and ‘longitude’ using reciprocal transformer, 'calculated\_host\_listings\_count', 'reviews\_per\_month', 'availability\_365' and 'minimum\_nights' using YeoJohnson transformer, and lastly, ‘availability\_365’ using power transformer using square root and 'reviews\_per\_month' and 'number\_of\_reviews' using power transformer using cube root.

The power transformer was already explained during the HR analytics section, therefore, I will instead explain reciprocal transformer and Yeo Johnson transformer.

The reciprocal transformer is a feature transformation technique to modify numerical

features by taking their reciprocal or the inverse of each value (1/x). This is useful when the feature before transformation has a skewed distribution with a large range of values to make it more symmetric and normalize it to an extent. However, the transformation cannot be applied when values contain 0 or negative values as it will result in negative reciprocal values and can lead the model to misinterpret the data during its analysis.

The Yeo Johnson transformer is another power transformation technique and is similar to the Box-Cox transformation. The Yeo Johnson transformation involves raising the numerical feature values x to a power of λ. The formula for the transformation is as shown in the image below.



The Yeo Johnson transformer is also able to handle positive, negative and 0 values

and aims to make the data more symmetrical and therefore can be beneficial for modeling techniques.

Since there were many features that were transformed I will show the before and after of ‘reviews\_per\_month’ and ‘number\_of\_reviews’ as they had the most desirable transformation.

‘reviews\_per\_month’:

Before:

A graph with a line and a red line

Description automatically generated

After:

A graph with a red line and a line

Description automatically generated

‘number\_of\_reviews’:

Before:

A blue line and red line

Description automatically generated

After:

A graph with a red line and a blue line

Description automatically generated

However, the transformation with the least effect was ‘minimum\_nights’ where there

seems to have no visible effect on the distribution.

Before:

A graph with a line going up

Description automatically generated

After:

A graph with a line

Description automatically generated

**3.3.4 Encoding categorical features**

The categorical features that need encoding are ‘host\_name’, ‘neighbourhood\_group’, ‘room\_type’, and ‘neighbourhood’, each with their own unique traits and therefore, different methods used to encode them.

First is ‘host\_name’, during the data exploration stage I explained that ‘host\_name’ had numerous different host names and I planned on rare encoding the feature since the rare values are very likely to be noisy and uninformative.

I made the threshold for rare categories 1% and that gave me a total of 10 different categories not including the rare categories as shown below. I then mapped the rest of the categories to ‘Rare’.

A screenshot of a computer

Description automatically generated

I then encoded ‘host\_name’ with target encoding to help capture the

relationship between categories and the target variable. I considered using frequency encoding since it is frequently used after rare encoding but with only 290 rows of data on the least rare category out of 7902 rows of data, the frequency of categories will be too low to capture its importance in the model.

Next is ‘neighbourhood\_group’ and ‘room\_type’ where I used one-hot encoding for the two features. I decided to do so as ‘neighbourhood\_group’ only consisted of 5 categories and ‘room\_type’ only consisted of 3 categories. The features’ low cardinality combined with the fact that we are using a linear regression model, using one-hot encoding is a very good option.

This is because this encoding method does not introduce ordinality and allows the linear model to treat each category independently without assuming a linear relationship between the categories. The low cardinality also results in the additional columns being manageable.

Lastly is ‘neighbourhood’, which consists of 43 categories. I considered using either target encoding or frequency encoding but due to the high number of categories target encoding has a higher risk of overfitting than frequency encoding.

**3.3.5 Scaling the data**

For scaling the data, I once again used the standard scaler for its reliability.

Below is the before and after effect of the values.

A comparison of a graph

Description automatically generated

I then ran a linear regression model using the scaled train data to get a baseline train

and test RMSE and R^2 value. In the end, my train metrics are an RMSE of 255.94 and R^2 value of 0.070515 and my test metrics are an RMSE of 564,240,000 and an R^2 value of 0.13808.

The baseline model shows clear issue of overfitting due to the unusually high RMSE in test data and the discrepancy in R^2 values. The high RMSE value is unusually large which means that the model’s predictions are significantly off and it is a good indicator that the outliers are severely affecting the model. This will be addressed in the improving the model section.

* 1. **Correlation Analysis**

In this section I once again plotted the “Pearson Correlation of Features” and a

scatter matrix.

The “Pearson Correlation of Features” show that there is not a lot correlation

between the predictor features and ‘price’ judging from the colors on the column/row of ‘price’. This suggests that there is a weak linear relationship between all the features and the target, and it may not be ideal for linear modeling. I do suspect that this is because of all the outliers within the target and predictor values contributing to this.

A screen shot of a computer

Description automatically generated

Looking at the scatter matrix, it is clear that there are a lot of outliers within ‘price’

with data points being very far from where most data points are clustered. From this it is also clear that many of the inaccuracies in the model is likely attributed to the outliers.

* 1. **Improving the model**

**3.5.1 Model Summary**

First, I created the model summary of the model that was run in the scaling the data

stage.

A table with numbers and letters

Description automatically generated

From the model summary it is clear that a lot of features do not have a good p-

values with most features having a p-value of over 0.1. Do take note that the p-values below is not definitive and p-values that are significant can vary based on model, dataset and problem context.

* p-value < 0.05: This is considered statistically significant. There's strong evidence that the feature has a real and non-random relationship with the outcome variable.
* 0.05 < p-value < 0.10: This is considered marginally significant. There's some evidence of a relationship, but it's not as strong as a p-value below 0.05. Further investigation might be warranted.
* p-value > 0.10: This is considered non-significant. There's not enough evidence to conclude that the feature has a real effect on the outcome variable. It's likely due to random chance.

With the p-values and their different thresholds shown above, it is easy to identify

many features that are not significant in the model. This might mean that there is a lot of noise in the data and the features might not offer meaningful insights, making the model more complex without contributing to its predictive power.

**3.5.2 Trimming outliers in ‘price’**

The first improvement that I made to the dataset was to remove ‘price’ that were

over 700 as well as took note of how many rows of data were lost in this process. I chose the price of 700 as I noticed that values higher than that were quite little in distribution and therefore decided that it would be easier to drop them. In this process 93 rows of data were dropped, leaving 7800 rows of data left. I then plotted the diagnostic plot of ‘price’ to look for how many more outliers are left in the target.

A graph with a line and a red line

Description automatically generated

There are still a lot of outliers in the target, but I was afraid of data loss. After this run

of the model I will experiment with forcibly remove the outliers.

The model’s performance using this dataset (after scaling) is, train metrics of

81.786 (Train RMSE) and 0.38658 (Train R^2 Value) and the test metrics 580,650,000. (Test RMSE) and 0.361065 (Test R^2 Value). The model is marginally better than that of the baseline with the test and train R^2 value going from 0.070515 and 0.13808 to the above values. Still, the R^2 values are still quite low which means that the model might not capture all the patterns or features that determine the prices accurately.

However, the RMSE is still in the hundred million for the test value. This high of an RMSE suggests strongly that the model’s predictions on the test are very far off from the actual values. This means that the model generalization and predictive capability does not perform well on unseen data.

This model has shown improvement, but it is still not satisfactory, meaning I will be

forcibly removing the outliers in ‘price’

**3.5.3 Removing outliers in ‘price’**

This time, to remove outliers in ‘price’ forcibly, I set the threshold to be 370 as after

A graph with a line going up

Description automatically generatedexperimentation I found this to be the best value to remove outliers but still retain model performance.

Running this model, I get the following train and test metrics.

The train metrics are 59.158 (Train RMSE) and 0.47222 (Train R^2 Value) and the test metrics are 364,610,000 (Test RMSE) and 0.46827 (Test R^2 Value). This is a proportionally good improvement from the previous run of the model.

This substantial reduction in RMSE and increase in R^2 value for both train and test sets indicate that forcibly removing the outliers has positively impacted the model’s performance. However, the RMSE in test is still substantially large which means that the model’s performance is still unsatisfactory.

**3.5.4 Dropping of features**

The next model uses the model summary to try and drop columns that are insignificant using the p-value. This time I created the summary table again:

A screenshot of a table

Description automatically generated

Comparing this summary table with the previous one that we created, there are a lot

more features that are significant after removing the outliers from ‘price’.

Using the above p-value threshold above the following features from the above were

dropped from the dataset:

['neighbourhood', 'latitude','longitude', 'calculated\_host\_listings\_count', 'month', 'day', 'year', 'neigh\_group\_Central Region', 'neigh\_group\_East Region', 'neigh\_group\_North Region', 'neigh\_group\_North-East Region', 'neigh\_group\_West Region']

Using this dataset, running the model gave me slightly worse results at train metrics of 59.795 (Train RMSE) and 0.46078 (Train R^2 Value). As well as test metrics of 467,700,000 (Test RMSE) and 0.45266 (Test R^2 Value).

Comparing this with the previous model, the previous model was better with a lower RMSE and better R^2 value. This might be due to the loss of data that were somewhat significant still in the model despite its higher p-value.

**3.5.5 Implementing PCA**

To overcome the noise in the dataset in hopes to achieve a higher model performance (which indicates better dataset), I decided to use PCA. As explained in the section in improving the model in the HR analytics, Principal Component Analysis (PCA) is a common dimensionality reduction technique that is more commonly used in data analysis and machine learning to simply the complexity of high-dimensional datasets while retaining important information.

In this use case, I decided to use PCA for its characteristic of it retaining important information.

In this model I used the data from after forcibly removing the outliers but before the dropping of features. After splitting the data from X and Y, running the PCA, the following shows the components of the new data as well as their variance.

A graph with a line

Description automatically generated

I also printed the features and their respective variance explanations:

A number on a white background

Description automatically generated

The explained variances might be hard to tell but I used these values to experiment

with the cumulative variance threshold and found that a 100% variance was the best for model performance. By selecting 100% of the variance, I have effectively preserved the entirety of the original information.



The above shows the number of components in that make up 100% variance. Notice that it is only 20 components in contrary to an above image that shows that there are 20 components. This discrepancy between the maximum number of components and the number of components that is required to capture 100% variance may stem from the fact that, only a subset of components is necessary to capture the entire variance of the dataset. In this case, PCA helped to identify the most critical components that retains the dataset’s key information.

The model performance has improved by a lot with a train metric of 59.158 (Train RMSE) and 0.47222 (Train R^2 Value) and a test metric of 69.655 (Test RMSE) and 0.46827 (Test R^2 Value).

This is a model performance is a lot better than compared to all the previous attempts at improving the model. The R^2 values for both train and test have improved slightly and the RMSE of train improved a bit but the RMSE of test improved drastically.

In previous attempts it was at hundreds of millions but this current iteration it is now at only 70. This is a major advancement, indicating that there is a substantial improvement in the model’s ability to generalize and make accurate predictions on new, unseen data.

With this I concluded the Airbnb modelling and exported the data of the model

that had PCA done on it with a 100% variance. It is saved as ‘lis\_export.csv’.

1. **Summary & Further Improvements**

**4.1 Summary**

In this report we have explored, cleaned, and transformed the data for both the HR Analytics dataset and Airbnb Listings dataset. I explored both datasets in detail uncovering the various nuances and problems that needed to be handled before the data is ready to be modelled. The report documents detailed steps of examining feature distributions, handling null values, addressing outliers, transforming numerical features, encoding categorical variables, and improving the model.

In HR Analytics, I introduced the HR analytics dataset and defined the problem: predicting employee promotions based on various HR-related features. During the data exploration stage, I explored the imbalance counts of ‘is\_promoted’ and performed down sampling, as well as explored feature counts, data types, and null values.

I also went through a detailed exploration and insights into ‘education’ and ‘previous\_year\_rating’ features with null values, identified outliers in ‘age’ and ‘length\_of\_service’ and discussed their potential handling methods. I explain the purpose of the Q-Q plot in identifying normal distribution and its significance for transforming features. I also use a chi-square test to assess the relationship between ‘is\_promoted’ and ‘recruitment\_channel’.

During data cleansing and transformation, I have also outlined the steps for handling null values, managing outliers in ‘age’, transforming features ‘length\_of\_service’, ‘age’, and ‘age\_training\_score,’ and encoding categorical features as well as my rationale on why and how I handled each of the features.

I also explained the application of a standard scaler for data normalization and reviewed Peason Correlation and scatter matrix in order to observe relationships between features and ‘is\_promoted’

During the improving the model stage, I discussed outlier handling in ‘length\_of\_service’ using winsorization and its impact on model accuracy. I have experimented with feature removal and PCA for dimensionality reduction to improve model performance. Lastly I gave a brief summary on the model performance variations with changes in feature handling.

In Airbnb Listings, I begin with explaining the objective which is to build a regression model to predict ‘price’. I begin with data exploration to delve into various facets of the dataset. This is done by exploring the features, distributions, outliers, null values, and categorical variables. I also debated whether if I wanted to focus on a subset of data and decided not to as by doing so will result in a lot of loss of data.

During the data cleansing and transformation, I employed multiple strategies to understand and preprocess the dataset, imputing null values, handling outliers, transforming numerical features, encoding categorical variables, and scaling the data. I included detailed explanations on which transformers and which encoders to use on the features.

Meticulous attention was paid to understand the impact of outliers on model performance. Initial attempts at model building showcased overfitting and low predictive capabilities, primarily attributed to outliers within the target variable, ‘price’. Several iterations were made to address outliers in ‘price’ and significant features, coupled with experimenting with different transformation techniques and encoding strategies.

The final iteration, incorporating PCA, yielded a notable enhancement in performance, showcasing significant improvements in RMSE and R^2 values in the test dataset.

Overall, the report showcases a thorough data preprocessing, outlier handling, feature engineering, and model evaluation in building an accurate model to the best of my abilities. The two exported models, ‘hr\_export.csv’ and ‘lis\_export.csv’ is the result of these efforts.

* 1. **Further Improvements**

For further improvements, more advanced feature engineering techniques such as

interaction terms, polynomial features, or embedding categorical variables to capture more nuanced relationships within the data. Experimenting more advanced outlier handling techniques such as clustering-based outlier detection to see their impact on model performance could prove to be insightful as well.

Incorporating external datasets could also help to enrich the existing datasets and potentially improve predictive performance.

In the Airbnb listings dataset, exploring geographical data analysis such as deriving additional features from altitude and longitude such as distance from popular landmarks or clustering of locations could provide more insightful data for the model.