Predicting Prices for Airbnb listings in Paris and Overall Guest Satisfaction for new listings

# INTODUCTION AND PROBLEM

This dataset provides information on Airbnb prices in Paris, including room type, cleanliness rating, guest satisfaction score, number of bedrooms, and distance from the city centre. The attributes used in the dataset are room types, cleanliness and satisfaction ratings, bedrooms, distance from the city centre, and more to capture an in-depth understanding of Airbnb prices on weekdays. Using different methods, we analyse and identify the determinants of Airbnb prices across this city. Our dataset includes information such as real Sum (the total price of the listing), room type (private/shared/entire home/apt), host is superhost (boolean value indicating if host is a superhost or not), multi (indicator whether listing is for multiple rooms or not), biz (business indicator) , guest satisfaction overall (overall rating from guests comparing all listings offered by host ), bedrooms, dist (distance from city center) , lng & lat. The Paris data set offers insight into how global markets are affected by social dynamics and geographical factors which in turn determine pricing strategies for optimal profitability and the customer satisfaction in Airbnb. [1]

We want to predict: A. The price of a new listing, and B. The Overall Guest Satisfaction level of this new listing. If AirBnb could predict the overall satisfaction of a new listing, they could promote this listing even before getting reviews from the users.

On the other hand, price listing should be in given price range according to their characteristics (number of beds, satisfaction level, how close it is to the city, etc.). If a given listing is too far away from its predicted value, they could know beforehand if this price is in the correct range and suggest the users to increase or decrease the price for a better performance in the AirBnb website.

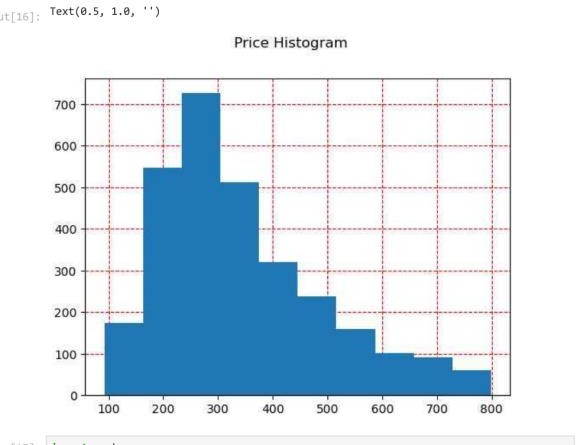
## DATA CLEANING AND PREPARATION

Initially we imported the data using panda’s library and then we used the command df. shape to see the different variables and records and we found out that we had 20 variables and 3130 records. Then we print the initial records in the data using the df. head command. Now coming on to cleaning and understanding the data, the first thing that we did was eliminating the first column and the two normalization columns from the data taking the help from statology.org as the first column was of no use in our data. [2]

# RENAMING THE COLUMNS

After this we renamed the columns for ease of use and we renamed the columns that we needed in the dataset for our analysis which were realSum, dist, metro and attr\_index. Then we rounded the values for the columns to two decimal places. After this we quickly observed that the average price

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

per listing was 398.79 per week and the average person capacity per listing is 2.95. Most of the listings were single rooms which can be seen by “multi” Boolean variable. We can see that the cleanliness overall rating is very high, averaging 9.25 out of 10. We can see the distance to the city center is an average distance of 3.01 km and the distance to the metro is an average of 230 meters. [3]

# PRINGTING MISSING VALUES

Fig 1.1

# E. CORRELATION GRAPH

Fig 1.3

After this we printed the missing values for all the columns. Then we used the matplotlib library to create boxplots for the different columns like guest\_satisfaction, host\_is\_superhost, city\_distance. From the output we can see that there are very high outliers in the price variable, which has to be removed to avoid distortions.

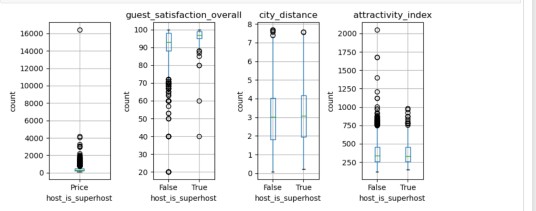


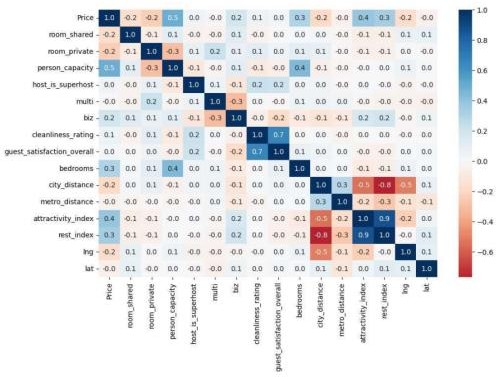
Fig 1.2

# DEALING WITH THE OUTLIERS

After this we ran a correlation graph to understand which were the most correlated variables to the outcome "Price" variable, and to find out if there was covariability between predictor variables and we found out that biggest correlation to the outcome "price" variables are: person capacity with "0.5" "attractivity\_index" with 0.4, "rest index" with 0.3 and "bedrooms" with 0.3. We drop city distance because this variable has a lower correlation to "Price".

We calculated the quartiles and interquartile range to determine outliers to later sum up the charges of the outliers. This way we can see how important outliers are for business. From the output we can see that there are 192 outliers out of 3130 records which means every price above 798.65 is

# F. BOX PLOT

Fig 1.4

considered a price outlier and the price outliers represent around 6.0% of the records. After this we removed the records with price over $1000 by taking the help from sparky examples. Then we removed the attractivity index over 1000 using the same sparky by examples.[4]

# CREATING HISTOGRAM

Then we created the price histogram, and we could see that now the price histogram is modified and has changed because of removing the outliers from the data.

Then we standardized the attractivity\_index predictor. The next step was to create a box plot and we saw that superhost prices had higher average price than non superhost listing but not necessarily with attractivity distance.

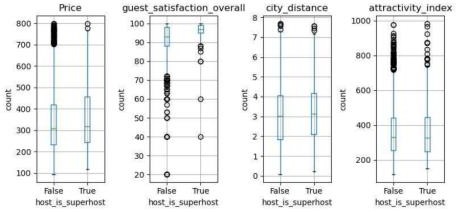
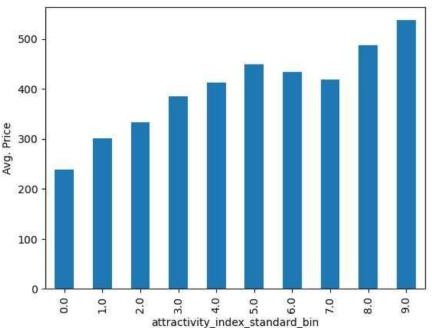


Fig 1.5

# G. CREATING BINS

Then we created bins for the predictors to improve performance and get more accuracy. We took help from geeks for geeks for binning the predicting variables. (Line 30) after binning and using “groupby” we could see that superhost listings were priced higher than non superhost. The bar chart for the average price and the attractivity index and city\_distance bin are as follows: [5]

 Fig 1.6

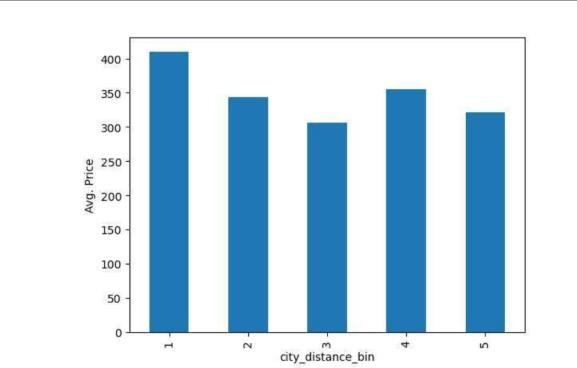


Fig 1.7

1. **BACKWARD ELIMINATION**

After cleaning and ordering the data set, we used a backward elimination process from a ChatGPT coding method, to identify the most relevant variables for the outcome “Price”. To do this we first had to make sure that categorical variables were transformed into dummy variables. We then proceeded with the backward elimination process where it excluded to predictor variables: 'guest satisfaction overall' and 'metro distance'. Looking at the confusion matrix chart we can see that both variables had a 0.0 correlation to price, so it makes sense that we should leave them out of the predictor variables. [6]

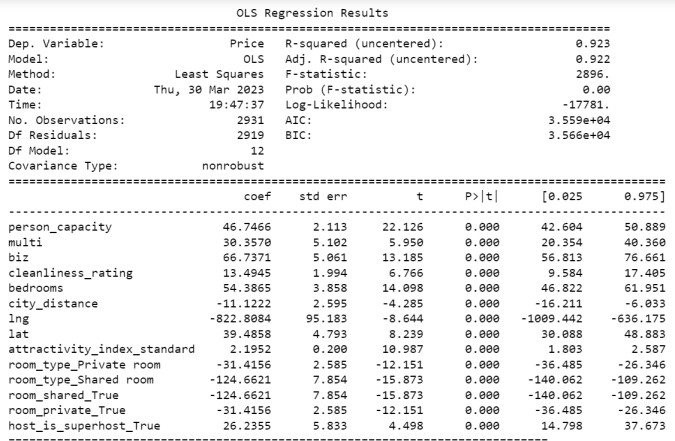


Fig 1.8

From the chart above we can see that all variables with a significance level less than 0.05, would positively contribute to the outcome variable. Furthermore, these variables count towards explaining 0.923 of the outcome variables, which is quite significant.

1. **1ST ALGORITHM: LINEAR REGRESSION**

Having defined our predictor variables, we proceeded with Linear Regression as our first algorithm. This algorithm is in fact the only one from the five algorithms we will use that will predict rather than classify the outcome variable. For the same reason, it’s the only algorithm that will predict the “Price” numerical variable, while the rest of the 4 algorithms used will classify the “Overall Satisfaction” categorical outcome.

To perform the linear regression model, we called on the needed libraries, split the data to training and validation data, created the linear regression model with the training data set and finally tested the performances of the training and validation sets. The training set gave us a mean percentage error of -8.94% and the validation set gave us a -7.88% mean percentage error, meaning that the model worked slightly better on the validation set than the training one.

We then proceeded to compare the residual analysis of both sets and found out their distribution is very similar and almost normally distributed.

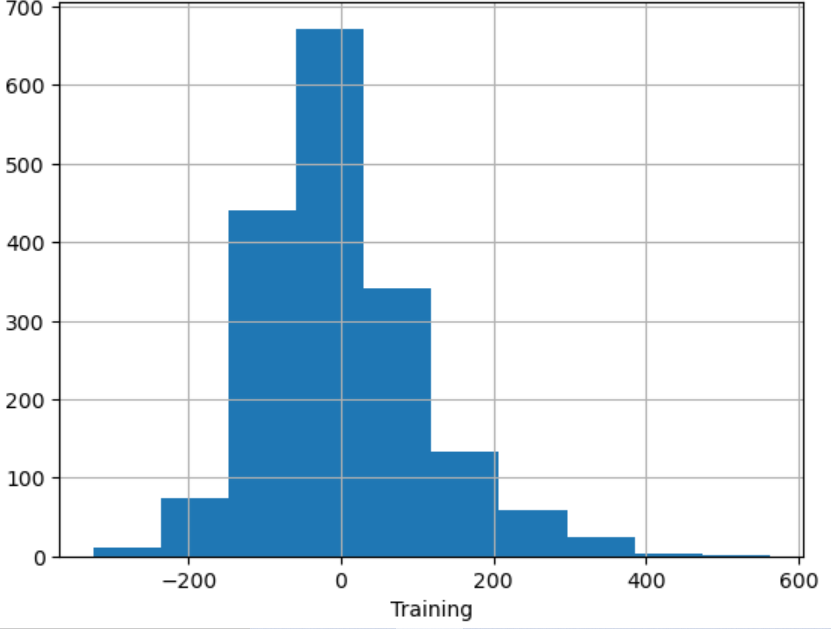


Fig 1.9

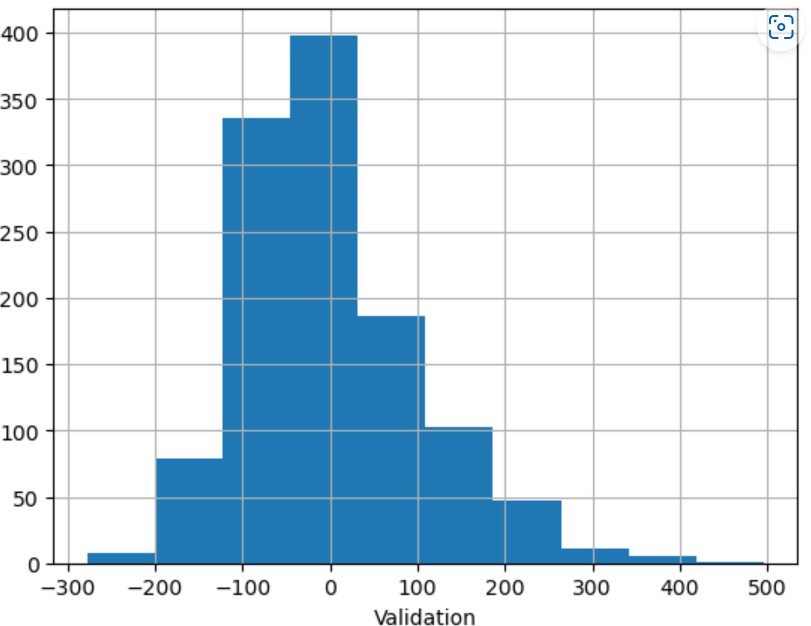


Fig 2.0

1. **2ND ALGORITHM: KNN CLASSIFIER**

We have used KNN CLASSIFIER to classify the overall satisfaction expected for a new listing. If the marketing team can predict the overall review of this listing, they could promote it on their webpage even before having reviews.

The new listing has a price of 530, distance from the city of

1.9 KM, a distance to the metro of 0.17KM, and cleanliness score of 8. We want to find out the expected classifier for this listing. For KNN we have filtered the dataset for predictors Price, city-distance, metro-distance, cleanliness-rating, and the target variable guest\_satisfaction\_overall. Since for KNN classifier the target variable should be categorical, we have binned guest\_satisfaction\_overall in to 5 bins as follows: Very Unsatisfied, Unsatisfied, Neutral, Satisfied, Very Satisfied.Then we split the dataset in to training and validation

sets. The training set accounts for 60% and the validation set accounts for 40% of the total. Then we created a model to classify the guest satisfaction for a price of 530, city distance of 1.9 km, metro distance of 0.17 km and a cleanliness score of 8.



Fig 2.1

We transform the entire dataset by normalizing to make everything to the same scale as our features are on different metrics. Then we use K nearest neighbor algorithm against the normalized training data with k =9. With K=9 it is expected that the classification of this new listing would be "Very Satisfied" in "overal1 guest satisfaction". But we still don't know which the best K is to use. Then we split data into training and validation sets. After that we train the classifier for different values of K and we choose K = 10 based on the accuracy of the model [7]. Then we rerun the algorithm on the combined training and testing sets to generate the classification of the new record.

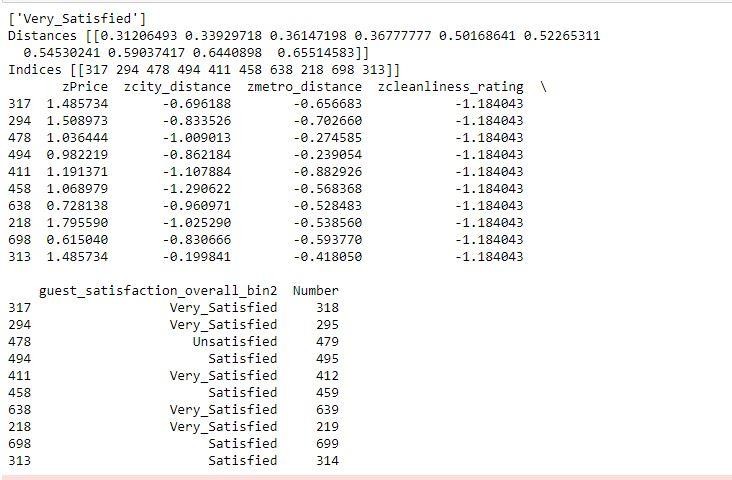


Fig 2.2

Thus, we conclude that a new listing with a Price of 530, distance from the city of 1.9 KM, a distance to the metro of 0.17KM, and cleanliness score of 8 would have an expected classifier of "Very Satisfied" in overall guest satisfaction.

1. **3RD ALGORITHM: CART**

Using the same new listing as in KNN classifier method, we want to test if a Classification and Regression Tree would also classify this listing as a “Very Satisfied” guest. We have used the same dataset as used in KNN. Here we dropped the number and guest satisfaction overall columns as they are not important for our analysis. Then we split the data in to training and testing sets. The training set accounts for 60% and the testing set accounts for 40% of the total.

Then we had done the CART model on binned guest satisfaction with a depth of 3 nodes.

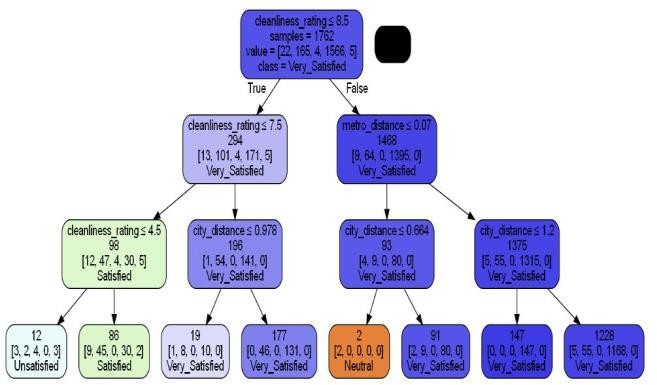


Fig 2.3

Thus, we found that for our new listing with: Price=530, city-distance=1.9KM, metro-distance=0.17KM, cleanliness score=8, this regression tree with a 3-depth level would classify as a "Very Satisfied" guest. The same value classification as in KNN method. Then we checked the accuracy for the training and validation data at a regression table of level of 3. We found that the accuracy score is 90.06% for training set and 90.13% for the validation set.[8]

We then continued trying different depth levels for the regression tree. At level 5 we increased accuracy of training data to 91.1%, but validation data dropped to 89.2%. At level

10 we improved accuracy of training data to 95.3% and decreased accuracy of validation data to 87.9%. This trend is caused by the over fitness of the model, where increasing the levels of the regression trees improves the accuracy of the training data but deteriorates the accuracy of the validation data. Due to this we decided that level 3 is an optimal level, where both training and validation data obtain a similar level of accuracy above 90%.[8]

1. **4TH ALGORITHM: NAÏVE BAYES CLASSIFIER**

After we try to classify guest satisfaction with KNN and CART using several predictors, we want to use more classifier algorithms to prove our result, so we run the algorithms of the Naïve Bayes Classifier and Logistic Regression with the same predictors.

Since some of our predictors such as guest satisfaction, price, and metro distance are numerical data, and we learned that the Naïve Bayes classifier can only deal with categorical predictors [9], we bin the numerical predictor and convert them to categorical data before splitting them to the training set and valid set.

After we fit the model with the training dataset, we test for accuracy with the valid dataset. We set up several conditions randomly, which is a very expensive room that has a very close distance to the metro and city center, and with an 8/10 rating of cleanliness, and compare the predicted results with actual results.

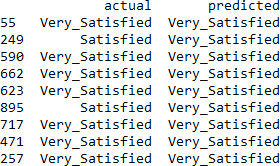


Fig 2.4

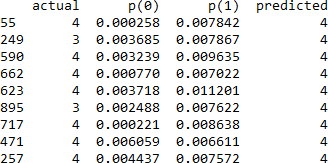
From the results above we can see that the accuracy of very satisfied bin is high, a new listing with price of 530, distance from the city of 1.9KM, a distance to the metro of 0.17, and cleanliness score of 8 would have an expected classifier of “Vary Satisfied”. This result is the same as the results in the KNN and CART models. [10]

However, we also notice that all predicted results are very satisfied, we have a concern about overfitting of results “very satisfied.”

1. **5TH ALGORITHM: LOGISTIC REGRESSION**

For our last algorithm, logistic regression, we also split the dataset to training set and valid set and fit the model with the training set.

To prove the accuracy of our model, we decide to use the same case as our 4th algorithm model, so we choose index 55,249,590,662,623,895,717,471 and 257 of the valid dataset as our test case, and it gives us the same result as previous model.



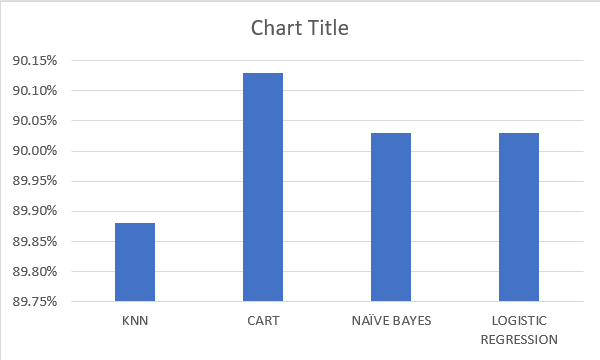
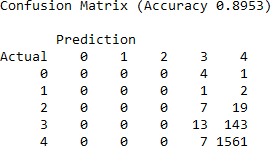
However, since we have a concern of over fitting, we calculate the confusion matrix for both testing set and valid set.

Fig 2.5

Fig 2.8

Here, the bar graph shows the accuracy of each model, and each model classified the “The Overall Guest Satisfaction” with these accuracy levels: KNN 89.88%, CART 90.13%, Naïve Bayes 90.03%, Logistic Regression 90.03%. [Note: All 4 models classified our new listing as "Very Satisfied" (which is a 4 out of 5 scale)]

**II. CONCLUSION**

Fig 2.6

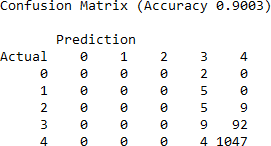


Fig 2.7 From the confusion matrix, we can see that even the

accuracy is high for both testing set and valid set, which are 89.53% and 90.03 separately, but the accuracy to predict the low satisfaction room are low, the low satisfaction room are predicted as high satisfaction, we think there might be overfitting because the limitation of predictors.

# I. COMPARISON OF ACCURACY

Finally comparing the Accuracy of KNN, CART, Naïve Bayes, Logistic regression.

For the prediction Error, the linear regression model gave us a Mean Absolute Percentage Error (MAPE) of 26.2%, that’s a wide error created by the variance of the data.

After building different classification models to predict guest satisfaction we learned that all 4 models predicted “Very Satisfied” guest reviews, which is the highest possible review.

We were able to build these models by using variables we can have knowledge of like: distance from city center, distance from metro, price and cleanliness rate.

Secondly, we were also able to use a linear regression model to predict the weekly price of our new listing. However, due to the high variance in the data set we were able to achieve an accuracy score of 26%, even after eliminating outliers. In successive stages we could use this same data set with other predictor models to learn if we can improve this accuracy rate.

**REFERENCES**

1. L. (2023, March 7). Airbnb Prices in European Cities. Kaggle. https[://www.](http://www.kaggle.com/code/larsboertjes/airbnb-)kag[gle.com/code/larsboertjes/airbnb-](http://www.kaggle.com/code/larsboertjes/airbnb-) prices-in-european-cities/
2. Zach, “How to Drop First Column in Pandas DataFrame (3 Methods),” *Statology*, Oct. 27, 2021. https[://www.](http://www.statology.org/pandas-drop-first-column/)s[tato](http://www.statology.org/pandas-drop-first-column/)lo[gy.org/pandas-drop-first-column/](http://www.statology.org/pandas-drop-first-column/)
3. “python - Detect and exclude outliers in a pandas DataFrame,” *Stack Overflow*. https://stackoverflow.com/questions/23199796/detect-and- exclude-outliers-in-a-pandas-dataframe
4. NNK, “Pandas Drop Rows With Condition,” *Spark By*

*{Examples}*, Jan. 11, 2022.

https://sparkbyexamples.com/pandas/pandas-drop-rows- with-condition/#:~:text=Use%20pandas.

1. “floor() and ceil() function Python,” *GeeksforGeeks*, Jan. 10, 2018.

https[://www.](http://www.geeksforgeeks.org/floor-ceil-function-python/)gee[ksforgeeks.org/floor-ceil-function-python/](http://www.geeksforgeeks.org/floor-ceil-function-python/) [6]ChatGPT: Used on March 20, 2023. ChatGPT. https://chat.openai.com/chat

1. “How to print an entire Pandas DataFrame in Python?,” *GeeksforGeeks*, Mar. 25, 2021. [Online]. Available: https[://www.](http://www.geeksforgeeks.org/how-to-print-an-entire-)gee[ksforgeeks.org/how-to-print-an-entire-](http://www.geeksforgeeks.org/how-to-print-an-entire-) pandas-dataframe-in-python/
2. “sklearn. metrics.accuracy\_score,” *scikit-learn*. [Online]. Available:

<http://scikitlearn.org/stable/modules/generated/sklearn.metri> cs.accuracy\_score.html.

[Accessed: Apr. 03, 2023]

1. Week10 note page 3
2. Awan, A. A., & Navlani, A. (2018b, December 4). Naive Bayes Classification Tutorial using Scikit-learn. https:[//www](http://www.datacamp.com/tutorial/naive-bayes-scikit-learn).d[atacamp.com/tutorial/naive-bayes-scikit-learn](http://www.datacamp.com/tutorial/naive-bayes-scikit-learn)