Data Science Programming

Data preprocessing, modeling, and reporting

Dr. Bassam Alkasasbeh

Done by: Sama Al-Qaisi

***Table of Contents***

[**Technical Documentation** 2](#_Toc126150550)

[**1** **Introduction** 2](#_Toc126150551)

[**1.1** **Types of data structures** 2](#_Toc126150552)

[**1.2** **Common libraries** 2](#_Toc126150556)

[**1.3** **Plotting and visualization libraries** 3](#_Toc126150557)

[**2** **Experiments** 3](#_Toc126150558)

[**2.1** **Programming languages and Tools** 3](#_Toc126150559)

[**2.2** **Load Data and Prepare Data (Pre-processing)** 3](#_Toc126150560)

[**2.3** **Approaches** 6](#_Toc126150561)

[**3** **Results** 6](#_Toc126150562)

[**3.1** **Compare the different models.** 6](#_Toc126150563)

[**3.2** **Charts** 7](#_Toc126150564)

[**3.3** **Analysis of the results** 8](#_Toc126150565)

[**4** **Evaluation** 9](#_Toc126150566)

[**4.1** **The choice of data structures** 9](#_Toc126150567)

[**4.2** **Selection of the appropriate libraries** 9](#_Toc126150568)

[**4.3** **The effectiveness of different models** 10](#_Toc126150569)

[**4.4** **Recommendations** 11](#_Toc126150570)

# **Technical Documentation**

# **Introduction**

## **Types of data structures**

## Data structures are the process of putting the data in order by sorting it and categorizing the data in an organized way, some of these structures allow modifying the data and manipulation, on the other hand, some of them will not allow doing so.

## Examples of data structures are Lists, Dictionaries, Tuples, and Sets. Some of them are known as Mutable which are the ones that can do the modifying and manipulation like lists, dictionaries, and sets, also some of them cannot like tuples which are known as Immutable. All the examples I have mentioned before are considered built-in data structures, but some data structures can be applied using libraries such as NumPy and Pandas which I will go into further in detail about later.

## The part of choosing which data structure to use depends on how to deal with your data, and what you want it to perform.

List: A type of data structure that is used to store elements in a specific order.

Dictionary: A type of data structure that stores values by putting a label on these values as a specific key to recognize values from each other according to their keys.

Set: A type of data structure used to store data without order and eliminate duplicates, allowing the items to appear once.

Tuple: A type of data structure that is used to store elements in a specific order, which also allows duplicates, but it does not allow the programmer to change the data or modify it.

## **Common libraries**

A library is a module that contains different classes with functions programmers can use by importing the library, after the library is imported the programmer will have the privilege to access the classes and functions of the library.

In python, most programmers are keen to use the NumPy and Pandas libraries because of the functions they provide. Also, in the process of using machine learning models the programmers keen to use the Scikit- learn library. And the definition of each library is as follows:

Numpy: Stands for Numerical Python library which is a library that has the function of building arrays and matrices, which can deal with single and multidimensional arrays, and what makes it special having functions that would be able to apply mathematical operations in these arrays and matrices. This library is most useful to be used by data scientists because they would have the ability to modify and process huge numerical data.

Pandas: It is a library that provides data frames as well as series which provides functions to manipulate and modify these data, which specializes in providing functions to help in the reprocessing stage of data and represent it as visuals.

Scikit-learn: It is a library that is known as sklearn which is used in the machine learning stage because of the functions, tools, and algorithms that will help in training, and establishing ML models for data sets.

## **Plotting and visualization libraries**

Data scientists need to use libraries that will conclude the data in visual models so it will help them analyze their data, and an example of the library that programmers use in python is the Matplotlib library.

which is as follows:

Matplotlib: A library that provides functions to create diverse ways to visualize your data such as box plots, bar charts, and radar charts. it contains functions that can help you to customize these charts as you want, and you can add features for them for example changing the color and adding legends.

# **Experiments**

## **Programming languages and tool**

Programming languages are some procedures and instructions the developer writes that the computer will work on executing, there are many programming languages such as Python. To implement these languages, you need a space to write in and a tool to write with, these tools are available for programmers to use such as Colab.

Python: Python is a language that is known for its simple syntax and ease to read as well as the variety of libraries that can be provided, data scientists are keen to use it because it can be used for data analysis, machine learning, and visualization of data, python considered as a high-level-language but it can be a great language for beginners because of its simplicity.

Colab: This is a programming tool that is also known as Google Colaboratory, it is a Jupyter environment that runs free on the cloud, and programmers use it as a space to write their code using programming languages such as Python and run it as well as analyze data with visualizing it, also providing the ability to run and build machine learning models, data scientist keen to use it to hold data science projects.

## **Load Data and Prepare Data (Pre-processing)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Column/step Name** | **Description** | **Justification** |
|  | **CASE\_ID** | Drop the column CASE\_ID | This column didn’t contain any null value, but I had to drop it because not having it won’t affect any column and the relation with the other columns is weak. |
|  | **RESOLUTION**  **RESOLUTION\_DESCRIPTION**  **CASE\_DESC** | Drop the columns.  RESOLUTION  RESOLUTION\_DESCRIPTION  CASE\_DESC | I have dropped these three columns because they did not depend on other columns and if they did, it will be a very weak connection that won’t affect the data set if they are gone, also these columns have a huge number of nulls, and so deleting them will be more effective than filling these nulls values. |
|  | **OFFER\_NAME** | Split the product column into two groups which will also split the offer name according to the product and fill each group with the mode of the group, which will fill the nulls in the offer name depending on the product column.  Using group by method. | I have concluded that the offer name is related to the product column so when the product is internet there will be certain attributes in the offer name, and when the product is mobile there will be certain attributes related to it because of my finding, I had to split the product and fill the offer name according to it. |
|  | **CUSTOMER\_GROUP** | Split the customer type column into two groups which will also split the customer group according to the type and fill each group with the mode of the group, which will fill the nulls in the customer group depending on the customer type column.  Using group by method. | I have concluded that the customer group is related to the customer type column so when the type is CBU there will be certain attributes in the customer group, and when the type is EBU there will be certain attributes related to it because of my finding, I had to split the customer type and fill the customer group according to it. |
|  | **ESCALATED\_GROUP** | Fill the escalated group column with no group when the escalation flag is no but if it was yes, the escalated group will be kept as it is, which the escalated group will be filled depending on the escalation flag, but after this, the null values will decrease to 42, so I filled them with the mode of the escalated group when the escalation flag will be yes. | I have done this to the escalated group because I have noticed that the escalated group is connected with an escalation flag if the escalated flag is yes the complaint had been escalated to the back office group, so if the escalated flag is no the complaint won’t be escalated to the back office group that is why I have put no group, and the rest was filled with the mode because there a very few of them that it won’t have a huge difference. |
|  | **CLOSE\_USER & OPEN\_USER** | Fill the close user column with open user value when the escalation flag is no but if it was yes the close user will be kept as it is, which the close user will be filled depending on the escalation flag, but the number of close user column will decrease to 447, so to decrease the number of nulls the close user with open user value when the open user is as close user column if it was yes the close user will be kept as it is, which will decrease the number of nulls to 362, so to fill the rest of the nulls, the nulls in the open user are the same as the 362 null values in the close user, which in this case I can’t connect them or with any other column so I filled them with no user. | When the escalation flag is no the open user is the same as the close user that is why I filled the close user as the open user when it is no but if it was yes the close user is not the same as the open user, also there won’t be a close user if there is not an open user, that is why when there is no open user but there is a close user, the close user will be filled according to the open user, then I have reached that the nulls in the open user are as the same as the close user, I decided to put no user for each because I can’t fill them according to any column. |
|  | **CLOSE\_GROUP** | Fill the close group with no escalation when the escalation flag is no but if it was yes, the closer group will keep as it is, and the close group will be filled depending on the escalation flag. | I have concluded that the close group is related to the escalation flag so when the escalation flag is no means the complaint hadn’t been escalated to the back-office group there won’t be any escalation in the close group but if the escalation flag is yes there will be a close group, so I kept it as it is. |
|  | **CLOSE\_DATE** | Fill the close date with no date when the current status column is active but if it was yes, the close date will be kept as it is, and the close date will be filled depending on the current status. | I have concluded that the close date is related to the current state, because if the current state is active means the complaint is still in process that is why there is no close data because it has not closed. |
|  | **AGE\_BRACKET** | Fill the age bracket with -1 | The -1 is an indication that the phone call or case has been opened and closed on the same day or the day before. |
|  | **CALLBACK\_MECHANISM** | Fill in the callback mechanism with the phone when the age bracket is one, but if it was not, the callback mechanism will be filled with the mode, which will be filled depending on the age bracket. | I have concluded that the callback mechanism depends on the age bracket, I have decided to fill it with the phone because the age bracket is 1 which means it had been opened and closed in a brief time which means the communication channel to contact the customer back with is logically had been the phone, and if it wasn’t 1 it will be filled with the mode because most of the communication channels are SMS. |
|  | **OPEN\_GR** | Fill the open group with no open group. | I have filled it with no open group because I did not see any relation with another group and the column, I will do training on is the product column which will not affect this column. |

## **Approaches**

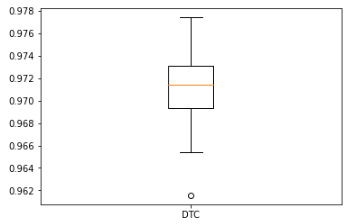
|  |  |  |
| --- | --- | --- |
| **Approach no.** | **Name** | **Description** |
|  | **KNN** (K-Nearest Neighbors) | A machine learning model that focuses on finding the k number of the nearest data to another data, by finding the distance between the points, the accuracy of this approach depends on the k-parameter that the user defines at the beginning of the algorithm, and the k I have used is 3 and this model considered a supervised ML model because the output for each input is known, also KNN can’t work with categorial data, just with numeric data. |
|  | **DT** (Decision Tree) | A machine learning model that focuses on splitting the data into small groups like a tree based on the values of the chosen feature, this ML model can be used to make predictions on the new data, this ML model can manage both categorical and numerical data. |
|  | **GNB** (Gaussian Naive Bayes) | A machine learning model that focuses on working with numeric data because it follows the Gaussian distribution, by finding the mean and standard deviation for each feature and deciding to which they belong, GNB is known for working on continuous data rather than discrete. |

# **Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approach no.** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|  | 0.8476854076565752 | 0.7970455786916459 | 0.6624060235942927 | 0.7233520089151884 |
|  | 0.971600192215281 | 0.9529705579739842 | 0.9526946853022482 | 0.9527776099391679 |
|  | 0.8323722569277592 | 0.6911597540870886 | 0.8001863721734102 | 0.7415406773971055 |

## **Compare the different models.**

## **Charts**

Chart, box and whisker chart

Description automatically generated

**Box-plot chart viewing the accuracy in each iteration for each Machine Learning model.**

Chart, radar chart

Description automatically generated

Chart, bar chart

Description automatically generated

**Bar chart viewing the average of the three performance measures (precision, recall, f1\_score) for each Machine Learning model.**

**Radar chart viewing the average of the four performance measures (Accuracy, Precision, Recall, F1\_score) for each Machine Learning model.**

## **Analysis of the results**

After the machine learning models stage followed by the visualization stage, they must be followed with the result analysis step because it will help in getting insights into how the performance of the model was whether it performed well or the areas that underperformed.

The results that have been produced from my project were from the three machine learning models that I have applied to my dataset which are the **KNN** (K-Nearest Neighbors) when K is three, **DT** (Decision Tree), and **GNB** (Gaussian Naive Bayes). These three approaches were needed to find the four machine learning measures that will represent the model’s performance: the f1 score, accuracy, precision, and recall. Then I have represented these results as a chart to read the result easier.

First, I will go through viewing the average that I have found for each model, as it was shown the decision tree model held the higher average of the four measures (accuracy, precision, recall, and f1-score) out of the three models by 0.97%, 0.95%, 0.95%, and 0.95%, but on the other hand, the GNB and KNN the readings between them were different to varying degrees, because the KNN hold a higher (accuracy, precision) than the GNB by 0.84%, 0.79%, but the GNB hold a higher ( f1 score, recall) than the KNN by 0.74%, 0.80%.

Second, I will go through the charts I have implemented.

The first was a 2 box plot (one for the DT and one for the KNN and GNB) that combined the accuracy of each iteration for the three machine-learning models, as is shown in the graph above the box plot provides us with information on the maximum, minimum value, first and third quartile, as well as the median for each one, as you can see each of the three models have a minimum and maximum value, which are different from each other, as you can see that the DT is higher than the KNN and GNB and the values are not widely distributed, as for the GNB you can observe that the data is skewed to the right, and the KNN has a wide range of observation, also if you look correctly for KNN and GNB there is not any outliers result which conclude in not having any unusual observations but the DT there is a result of an outlier.

The second was a bar chart that viewed the average of (f1 score, precision, and recall) for the three machine learning models, as is shown the blue bar is for the Decision Tree model which is higher in the 3 measures than the rest two models, and the green bar is for the KNN model which is higher in precision than the GNB, but the GNB is higher than KNN in the recall and f1 score which is the yellow bar.

The third was a radar chart that viewed the average of the four measures for each model, as shown the DT ( the orange line) got higher results for the four measures than the rest 2 models because the borders of the DT are the nearest ones for the upper circle, and the KNN and GNB are on top of each other and overlapping because the KNN (the blue line) model which is higher in precision than the GNB, but the GNB (the green line) is higher than KNN in the recall and f1 score, and as you can see sometimes the KNN will be the nearest to the smaller circle and other times is the GNB.

# **Evaluation**

## **The choice of data structures**

For my project, I have used different data structures and every one of them is used in the place that is needed, the most used data structures in my code were, lists, dictionaries, and data frames, the reason behind using them are as follows:

I have used the list when I needed to store values whether they were integers, float, strings, or objects, I have used lists because the list gives you the ability to modify and alter the content of the list, that is why it is considered as mutable so it allows the programmer to append value to the list and made changes on it, also I have used dictionaries while drawing the box plot because I wanted to store each collection of data under the label (key) which will result in a key-value shape, also the dictionary is a mutable data structure which will also allow me to make changes on the values. I Did not resort to using the tuple because it is considered immutable which will not support my code if I needed to make changes, also the set was not practical, using it will not give me the choice to do indexing if I wanted to, or to store my data in order.

Also in my code, I have used a data frame data structure because I needed to load the data set on the Colab environment to read it and work based on it, data frames are used also in my code because I needed to implement data cleaning and preprocessing that required to filling the missing values by using group by functions and using aggregated functions, also data frames had been used in the machine learning processes to split the data into two data frames and do the testing and training as well as analyzing the data after visualizing it.

## **Selection of the appropriate libraries**

The libraries I have resorted to implementing in my code are the NumPy, Pandas, Scikit-learn, and Matplotlib libraries, each one of which was used in the place where it was needed.

NumPy: This library was used to help me handle arrays and matrices with multidimensional arrays, the main use of this library was the mathematical functions that did provide for helping me handle missing values. which helped me in the cleaning and preprocessing step, where I use (where) the function that will not work without importing the Numpy library, also I used this library in the visualization stage where for example I had to concatenate two arrays.

Pandas: This library is an important library to implement my code because I had to use it to help me load data from a CSV file, also I used the data frame type of data structure that the library provides, also this library was used in the cleaning and handling missing values step where I had to use it for dropping duplicates and for using the tool that will help me like using the aggregating function, also I used it when I had to use the methods for calculating the mode statistic, this library was also used in the visualization step where I had to use the boxplot method that needed to import boxplot from the from (pandas. Plotting).

Scikit-learn: This was an essential library to use in my code used mainly in the machine learning stage generating the different machine learning models with their algorithms, for my project I have imported (KNeighborsClassifier) from (sklearn. neighbors), (DecisionTreeClassifier) from (sklearn. tree), and import (GaussianNB) from (sklearn.naive\_bayes ), also was used when I needed to split the data frame into training and testing frames, when I needed to evaluate the effectiveness of each model I needed to use the machine learning measures which are the (f1 score, accuracy, precision, and recall) I resorted to importing (accuracy\_score, precision\_score, recall\_score, f1\_score)from (sklearn. metrics). Also, in implementing the ML models I had to encode all the categorical data into numeric which I have done using label encoder by importing (LabelEncoder)from (sklearn. preprocessing), so in conclusion the Sklearn was a huge part in implementing my project.

Matplotlib: This library was mainly used in the visualization stage of my project, which I had to use it to build the charts that will help me analyze the performance of the machine learning models, so I have used it when I needed to customize the charts like adding and modifying axis labels, titles, and legend.

## **The effectiveness of different models**

I have applied 3 different machine learning models, and each of them solved the same problem I have at hand, then compared the results from each model in the analysis part, and now I will go deeper into each model and evaluate it depending on the effect they had on the problem.

The problem the ML models have worked on is working on the product label to give a proof of concept for the company, by generating the different 3 models that I have chosen.

The machine learning models that I have chosen are the KNN (K-Nearest Neighbours), the Decision Tree, and the GNB (Gaussian Naive Bayes). I will go through each one of them and present the reason for choosing them and what are the areas they considered strong choices, and the areas they limited their performance.

The Decision Tree was the model that held the higher results from the accuracy, precision, recall, and f1, and that was because the decision tree focuses on splitting the data into small groups like a tree, based on the values of the chosen label, it was a good choice because the decision tree can work with categorical and numerical data which was the case with our dataset, as well as that our dataset considered a large one, which makes it suitable for this model because it works better with large data to generate good results, also it can handle missing values and outlier in a dataset, and may not be affected by them.

The Gaussian Naive Bayes and the KNN both have discrepancies between results, that is because the GNB has higher results for the f1 score, and recall, than the KNN, and the KNN held higher results for the accuracy, and precision than the GNB, and the readings between them were different to varying degrees due to that the GNB considers the features in a data set to be independent which is not the case with our data set and can lead to low accuracy, and that is because our data have dependent features, also the structure of the GNB can be difficult to interpret as well as the prediction of the GNB is affected by the outliers and having missing values. Moreover, the thing that played a role in the KNN performance was the sensitivity of the model toward the outliers and missing values, although I didn’t have missing values because I have filled them, I have values that have been filled an unmeaningful for the data which can affect the accuracy of the prediction.

## **Recommendations**

After being done with implementing the work, we need to provide our work with some recommendations on how to improve our work which will improve performance and help with identifying the places that will need to be improved to enhance the performance.

As for my project, I will be providing you with some recommendations that if I had done earlier will change the result that I already got for the better way.

First, I will start with the preprocessing, for what I have done, I have focused on finding relations between some of the columns, and filled the ones that I didn’t find any relations between them with values that are vague or don’t have a meaning which just filled the null values, but if I want to improve I may have calculated the age bracket depending on the open-closed columns, I must have looked deeper in the meaning and relations behind the columns which will allow me to impute the missing values in a valuable and correct way, also another way of improvement was to fill the missing value using a machine learning model, in first I have tried to build the code but didn’t implemented it in the right way which I have gave up and gone to the alternative choice which is filling them based on relations, but If I have filled the missing values using an ML model I will get a good accuracy because I will have more accurate predictions, also it will help in finding multiple relationships between columns that simple column relationships may not be able to handle, also to not just focus on handling messing values, but also another improvement is to focus on handling the outliers and imbalanced data.

Second, I may have tried different values of k to see the change in results and choose the best one, rather than just choosing an odd number which is 3, as a way of improvement I should have used an odd number that is not very small and not very large, because the small number will result in having overfitting with the data, as for the large number will result in having underfitting, so I would use a balanced k value to enhance the performance of the model.

Third, I have split the data into a training set and a testing set with a test size of 0.2, which will result in having 0.8 for training and 0.2 for testing, as I way of improvement I will recommend using a test size of 0.3 because it will help in increasing the ability to handle outliers and improving the accuracy.

Fourth, I have used the matplotlib library to help me with the visualization process by providing me with the charts I have used as the box-plot, bar chart, and radar chart, but as a way of improvement, I will use another library that will help me in providing functions and features that the matplotlib library don’t have which will help in improving the quality of the charts I will be building and presenting.