

**UNIVERSITY OF SANTO TOMAS**  
**College of Information and Computing Sciences**  
Machine Learning (CSELEC2C): Laboratory Exercise 2

***Can You Determine Leads?***

**Submitted by:**

*Kristel Lenci C. Cruz*  
[kristellenci.cruz.cics@ust.edu.ph](mailto:kristellenci.cruz.cics@ust.edu.ph)  
3CSC

*Jerome Anthony P. Mangalus*  
[jeromeanthony.mangalus.cics@ust.edu.ph](mailto:jeromeanthony.mangalus.cics@ust.edu.ph)  
3CSC

## **I. INTRODUCTION**

This document will provide in-depth explanations of the results obtained by the group from the laboratory exercise that was done as well as the techniques employed to produce these results. Moreover, this document will have five sections: Introduction, Methodology, Experiments, Results & Analysis, and Conclusion & Recommendations.

The primary objective of this laboratory exercise was to identify and predict possible customers that will avail of ABC Supermarket's upcoming year-end promo. To do this, the group will use the provided *Excel* dataset to design a proper and efficient machine-learning model. The said dataset contains the following features:

- |                   |                       |
|-------------------|-----------------------|
| ● ID              | ● MntFruits           |
| ● YearBirth       | ● MntSweetProducts    |
| ● Education       | ● MntWines            |
| ● MatritalStatus  | ● MntGoldProds        |
| ● Income          | ● NumDealsPurchases   |
| ● Kidhome         | ● NumCatalogPurchases |
| ● Teenhome        | ● NumStorePurchases   |
| ● DtCustomer      | ● NumWebPurchases     |
| ● Recency         | ● NumWebVisitsMonth   |
| ● MntFishProducts | ● Complain            |
| ● MntMeatProducts | ● Response            |

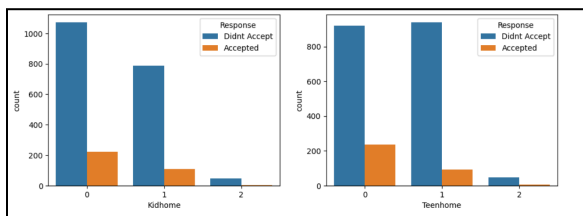
Upon initializing the dataset on *Jupyter Lab*, an open-source web-based development environment commonly used for *Python* programming, it was seen that the initial number of records consists of 2240 records with 22 columns, which are the mentioned features in the previous table.

## **Exploratory Data Analysis**

Prior to beginning the building of the machine learning model, exploratory data analysis must be used to examine the data thoroughly. This is done to determine which variables are necessary to determine the dependent and independent variables as well as the target variables.

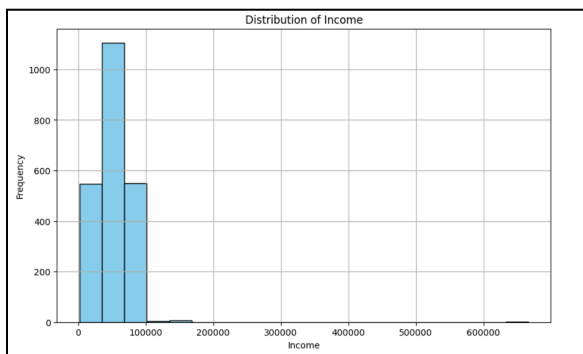


to analyze the responses of the said members visually. As shown in the graph, only 15% of the members will avail the upcoming promo, while 85% of the members will not be availing the said promo.



**Figure 1.5:** Bar Graph of Responses with Respect to Kid and Teen Home Features

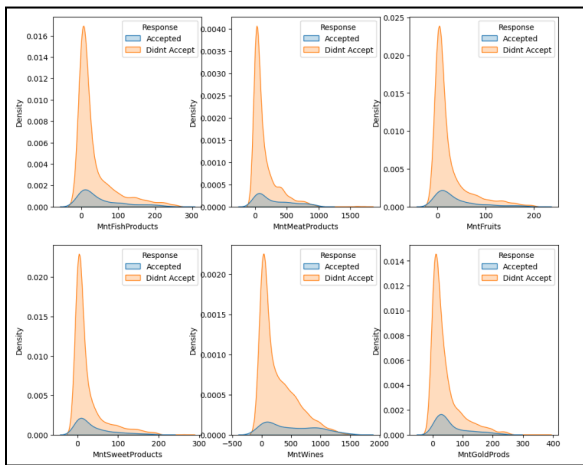
Figure 1.5 illustrates a bar graph representing responses related to the presence of children and teen home features. This graph holds significance as it may influence members' decisions regarding whether to avail of the promotional offer of ABC Supermarket. As depicted in the figure, members without children have predominantly availed the promo. The absence of children in the household could be a contributing factor to their choice to utilizing the promotional offer, possibly due to differing lifestyle priorities or preferences.



**Figure 1.6:** Bar Graph of Members Income

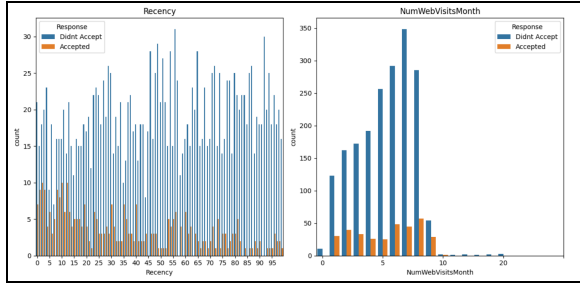
Figure 1.6 presents a bar graph illustrating the income distribution of all

members in the dataset. It is evident that there are outliers within the dataset, characterized by individuals with significantly higher incomes compared to others. In the following sections of this document, strategies for addressing these outliers will be explored and discussed.



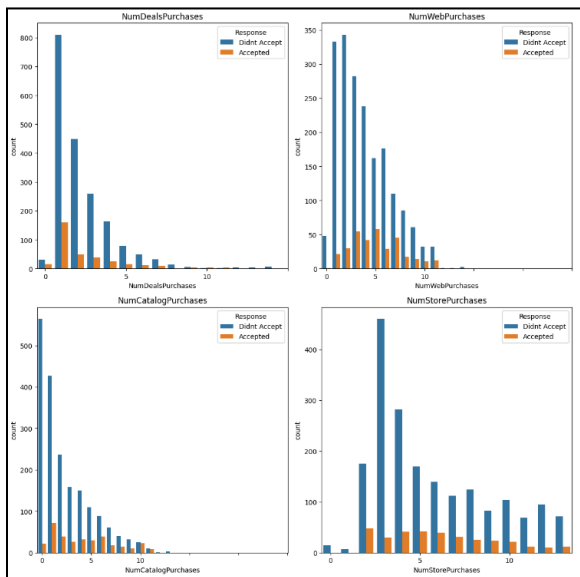
**Figure 1.7:** Kernel Density Estimation Plot of Products with respect to the Responses

Figure 1.7 shows the Kernel Density Estimation (KDE) of products purchased from the members with respect to their responses. This visualization tool explores the relationship between member purchases of various products and their responses, helping identify which products might have a stronger influence on customer response. However, the visualization suggests that the products purchased by members may not directly affect their responses regarding whether they will avail of the supermarket's upcoming promotion. This conclusion is drawn from the observation that the KDE plots for different product purchases exhibit similar patterns, indicating a lack of distinct correlation between product purchases and response decisions.



**Figure 1.8:** Bar Graph of Responses with respect to Recency and Website Visit

Figure 1.8 illustrates the relationship between member responses, their last purchase, and the number of website visits. These factors may contribute to members availing the upcoming promotion. The left bar graph displays the recency of purchases in relation to member responses. It is evident from this graph that a majority of members who availed the promotion made their last purchase recently. On the right, the bar graph demonstrates the relationship between member responses and their website visits. The data indicates that members who frequently visit the website are more likely to avail of the promotion.



**Figure 1.9:** Bar Graph of Number of Purchased Types

Four bar graphs comparing member responses across several variables are shown in Figure 1.9. We examine the connection between member's responses and the number of items purchased with a discount in the first graph. The purpose of this analysis is to determine whether members who purchase deals are more likely to take advantage of the promotion and maybe save money. Based on the said graph, users are more likely to take advantage of the promotion if they haven't bought many deals. The second graph looks at the connection between member responses and purchases made in the supermarket's website. It shows that some members have shopped online and have expressed interest in taking advantage of the upcoming promotion. The third and fourth graphs both analyze the relationship between member responses and two distinct types of purchases: catalog purchases and in-store purchases. Interestingly, the data shows that the number of purchases made through the catalog or in-store does not significantly impact member responses, as the acceptance rate of the promotion remains consistent regardless of the number purchased in-store or in a catalog.

## II. METHODOLOGY

After conducting a thorough analysis and profiling of the provided dataset, we have identified several variables with errors. To address these issues, we will clean the data and employ feature engineering techniques. The following sections will discuss how the said errors were handled:

### A. Year Birth

We choose to calculate the ages of individuals based on their birth year, instead of retaining the original 'Year\_Birth' column in the dataset. Doing the said procedure simplifies dataset interpretation, reduces dimensionality, and avoids potential data leakage issues. Figure 2.1 shows the updated values of ages.

```
Unique Ages:
[ 54 63 66 57 35 70 77 45 65 43 55 47 64 58 48 59 68 49
 53 38 52 50 34 37 40 56 69 41 51 46 72 62 60 42 61 67
 44 79 75 76 71 78 39 32 80 73 36 74 30 31 33 131 28 29
 125 81 83 84 124]
Total Number of Unique Records
59
```

**Figure 2.1:** Age Column Contents

### B. Membership Duration

The Dt\_Customer column in the dataset serves as a way to track the membership duration of each member in the dataset. However, as shown in Figure 2.2, most of the records for the said column were hard to interpret as they have different formats.

```
Dt_Customer: object
2240 elements with 663 unique elements
['6/16/14' '6/15/14' '5/13/14' datetime.datetime(2014, 11, 5, 0, 0)
datetime.datetime(2014, 8, 4, 0, 0) '3/17/14' '1/29/14' '1/18/14'
datetime.datetime(2014, 11, 1, 0, 0) '12/27/13'
datetime.datetime(2013, 9, 12, 0, 0) datetime.datetime(2013, 7, 12, 0, 0)
'10/16/13' datetime.datetime(2013, 5, 10, 0, 0)
datetime.datetime(2013, 11, 9, 0, 0) datetime.datetime(2013, 1, 8, 0, 0)
'7/23/13' datetime.datetime(2013, 1, 7, 0, 0) '5/28/13' '3/26/13'
'3/15/13' datetime.datetime(2013, 12, 2, 0, 0) '11/23/12' '10/13/12'
'9/14/12' '6/29/14' '5/31/14' '5/30/14' '4/27/14'
datetime.datetime(2014, 11, 4, 0, 0) '10/29/13'
datetime.datetime(2013, 9, 10, 0, 0) datetime.datetime(2013, 10, 5, 0, 0)
datetime.datetime(2013, 9, 5, 0, 0) '4/25/13' '4/20/13' '3/30/13'
datetime.datetime(2013, 1, 3, 0, 0) '2/14/13'
datetime.datetime(2013, 11, 1, 0, 0) datetime.datetime(2013, 3, 1, 0, 0)]
```

**Figure 2.2:** Uncleaned Dt\_Customer

To address this inconsistency, we opted to convert these to the most appropriate datatype, which is the datetime, and compute how long they have already been a member in years, similar in the Age

on the previous section. After cleaning the said column, we have reduced the total number of unique values from 663 unique elements down to 3 unique elements. These elements are 11, 12, and 13 years

### C. Marital Status

One of the first things we noticed while browsing the dataset is the invalid marital statuses present in some of the records in the dataset itself. Figure 2.2 shows the uncleaned marital status.

```
Marital_Status: object
2240 elements with 8 unique elements
['Divorced' 'Single' 'Married' 'Together' 'Widow' 'YOLO' 'Alone' 'Absurd']
```

**Figure 2.3:** Uncleaned Marital Status

To address these invalid entries, we converted them to the "Single" marital status. We made this assumption because some of the indicated statuses, such as "Absurd" or "YOLO," do not make sense. Additionally, even if a status like "Together" existed, we considered it to be single since it is not a recognized legal marital status. After cleaning, the unique values of marital status are as follows:

- Single
- Married
- Widow
- Divorced

### D. Null Values for Income

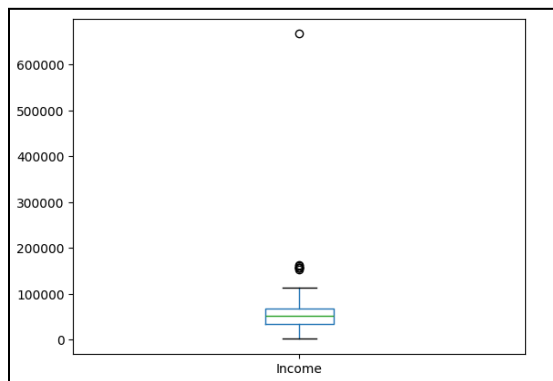
Referring to Figure 1.2 in the previous section of this paper, we can see that there are 24 null entries in the Income column. To handle this anomaly, we decided to calculate the average income across different Education and Marital Status. We decided to include education as one of the considerations in computing the average income since job opportunities may vary depending on the educational attainment of

one applicant. On the other hand, we included marital status because the description characterizes income as "household income." Therefore, we decided to include it, as household income may vary depending on an individual's marital status. We feel that this is the best way to fill in the null valued records because it is more specific to the degree the customer attained and the number of possible people that also contribute to the household's yearly income. Figure 2.4 represents the calculated average income in the dataset.

Education	Marital_Status	
2n Cycle	Divorced	49395.130435
	Married	46201.100000
	Single	48233.706522
	Widow	51392.200000
Basic	Divorced	9548.000000
	Married	21960.500000
	Single	19551.781250
	Widow	22123.000000
Graduation	Divorced	54526.042017
	Married	50800.258741
	Single	53714.529081
	Widow	54976.657143
Master	Divorced	50331.945946
	Married	53286.028986
	Single	52830.888268
	Widow	58401.545455
PhD	Divorced	53096.615385
	Married	58138.031579
	Single	54659.218605
	Widow	60288.083333

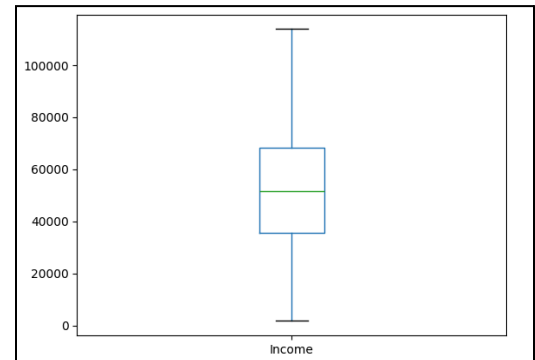
**Figure 2.4:** Average Income of the Dataset

### E. Income Outliers



**Figure 2.5:** Income Box Plot with Outliers

As shown in Figure 2.5, a few outliers are present in the Income column, which hinders the box plot from fully stretching. To address this anomaly, we have used Z-Score to eliminate these outliers. This is essential to do before feeding the dataset into the machine learning model, this assures us that this will produce robust and accurate models. Figure 2.6 shows the box plot after applying Z-Score in Outlier removal.



**Figure 2.6:** Income Box Plot without Outliers

### F. Products Purchased

As mentioned in the introductory section of this paper, the dataset consists of 22 columns in total, 6 columns contain the total number of a specific product purchased by a specific customer. These products are Fish, Meat, Fruits, Sweet, Wines, & Gold. Since these columns are not dependent on the target variable, and to reduce the number of columns the dataset has, we decided to merge these 6 columns into 1 column and named it "Total\_Mnt". After doing the said

approach, the columns of the dataset was reduced to 17 columns.

### G. Kid & Teen Home

Similarly, 2 columns in the dataset were used to represent the number of Kids and teens each member has. It was identified that these records does not affect the target variable, we opted to apply the same approach used in Total\_Mnt, and added a new column named “Num\_of\_Child”. After implementing the said approach, the dataset columns were reduced to 16.

ID	Education	Marital_Status	Income	Recency	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	Complain	Age	Membership_Age	Total_Mnt	Num_of_Child	
1	1224	Graduation	Unmarried	66250.0	0	1	4	4	0	1	0	34	0	1046	
2	1	Graduation	Single	370910.0	0	1	7	3	7	0	1	0	40	10	377
3	10478	Graduation	Married	472070.0	0	1	3	2	0	0	0	0	40	10	237
4	1046	Graduation	Single	340740.0	0	1	1	0	2	0	0	0	37	10	171
5	10771	Graduation	Single	274740.0	0	2	3	1	2	0	1	0	35	10	81
6	11	12	Unmarried	170	0	0	0	0	0	0	0	0	0	0	0
7	11420	MBA	Unmarried	440740.0	0	0	0	0	11	0	0	0	40	10	480
8	3203	2nd-Grade	Married	370300.0	0	1	1	0	0	0	0	0	47	10	330
9	3217	22	Graduation	Unmarried	462010.0	0	0	0	0	0	0	0	40	10	330
10	320	Graduation	Married	600700.0	0	1	5	4	10	0	0	0	40	10	1000
11	4270	MBA	Married	340770.0	0	1	0	0	0	0	0	0	35	10	1000

**Figure 2.7:** Dataset After Column Merging

### H. One Hot Encoding:

After cleaning the dataset, the next step is to convert categorical variables into a numerical one. This is done by using One Hot Encoding (OHE). The said transformation was utilized on Education and Marital Status.

### I. Training and Test Split

In training and test split, we separated the target variable, which was identified as the Response, from the selected features, which are ID, Income, Recency, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth, Complain, Age, Membership\_Age, Total\_Mnt, Num\_of\_Child, OHE of Education and Marital Status. The said separation will be stored on the x and Y variables, respectively. 70% of the data from both X and Y will be used for model training and 30% for testing.

Four variables that are produced by this process will be used in modeling. This procedure guarantees that the developed model operates effectively in predicting customers who are more likely to avail the promotion.

### J. Modeling

On this lab exercise, we were assigned to create a machine learning predictive model, which will predict whether the members in the dataset will avail the upcoming promotions or not. We are tasked with choosing one machine learning model from five given options. The choices include Logistic Regression, Decision Trees, Naive Bayes, K-Nearest Neighbor, and SVM. For our attempt in creating the said predictive model, we choose to use Logistic Regression. We also tried applying the other models given to us to see the differences of each model.

We decided to use logistic regression in creating the predictive model because this particular model is known to be suitable for binary classification problems and we are expected to identify the customers who are most likely to purchase the year-end offer.

### K. Hyperparameter Tuning

We utilized Grid Search for the hyperparameter tuning instead of the Randomized Search since this technique explores all possible combinations of hyperparameters from the parameter grid. On the other hand, the randomized search explores random combinations of hyperparameters wherein you define a probability distribution for each



hyperparameter, and the technique randomly samples a combination of hyperparameters. Although the grid search can be computationally expensive, we still chose this technique because there are not so many parameter grids in our dataset and model; hence, we decided to stick with this and get the best possible hyperparameters to improve our model.

```
Best hyperparameters: {'C': 0.001, 'max_iter': 100, 'penalty': 'l2', 'solver': 'liblinear'}
```

**Figure 2.8:** Results of Grid Search for the best Hyperparameters

### III. EXPERIMENTS

#### A. Removal of outliers in income column

Figures 2.5 and 2.6 show the boxplot of the income column that summarizes the said column. This figure represents the difference between the distribution of the data under the income column. We can clearly see that the distribution of data is much more ‘balanced’ and closer to one another, and we believe that this will positively affect the model in terms of its performance because it suggests that there is not a significant disparity in income levels among the data points. This balance could lead to a more stable and reliable model, as extreme outliers can sometimes negatively impact the model's ability to generalize well to new data. By having a more balanced distribution of income levels, the model may be better equipped to make predictions for a broader range of income levels,

potentially leading to more accurate results.

#### B. Creating an Age Group

As mentioned earlier, we converted and calculated the ‘Year\_Birth’ to the customer’s age; we tried two different approaches for this feature. First, we utilized binning in grouping the age into three categories: children, working, and elderly and dropped the age column. We thought that categorizing the age into these would be the most fitting for the One-Hot Encoding. Our second approach was not categorizing the age and keeping the age column. The first approach resulted in a much better-performing model since grouping them in a categorical manner most likely can lead to information loss, as the precise values of the continuous variable are no longer used in the model.

#### C. Results for Different Models

This section will briefly describe why we did not choose the model and the results generated from the pre-processing method we used for the different models: Decision Tree, K-Nearest Neighbor, and Support Vector Machine. The same data cleaning and feature engineering methods were used for this model to ensure that we can compare and contrast each model to the Logistic Regression model we have chosen. Hyperparameter tuning was also applied for all the models except for



the support vector machine model due to the lengthy computation time.

### Decision Trees

Decision trees may be useful for capturing non-linear relationships between features and the target variable due to it being straightforward, and this type of model can indicate which features are important in predicting the outcome. However, we did not choose this model as we have researched that this is prone to overfitting, and the dataset provided to us had many features.

```
Decision Tree Model
Accuracy: 0.8493
Precision: 0.4944
Recall: 0.4400
F1: 0.4656
AUC: 0.6805
```

**Figure 3.1 Decision Tree Model**

### Naive Bayes

The Naive Bayes model assumes that features are independent, which is not true for the dataset provided in this lab exercise. For instance, the probability of the customer purchasing the year-end offer might not be dependent on multiple features or might not be influenced by several factors. Furthermore, this model is often used with categorical features and can be affected by the complex distribution of the dataset; we decided not to use this model as the features in the data points are primarily numerical.

```
Naive Bayes Model
Accuracy: 0.7701
Precision: 0.3393
Recall: 0.5700
F1: 0.4254
AUC: 0.6876
```

**Figure 3.2 Naive Bayes Model**

### K-Nearest Neighbors

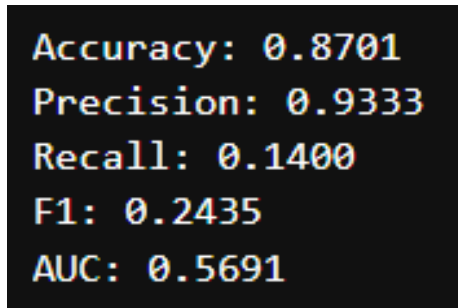
K-Nearest Neighbors is a simple model that works well with a small dataset. However, this is computationally expensive as the model computes the distance for each data point, and this considers all features equally important, which can negatively impact the model's performance.

```
K-Nearest Neighbor Model
Accuracy: 0.8567
Precision: 0.6000
Recall: 0.1200
F1: 0.2000
AUC: 0.5530
```

**Figure 3.3 K-Nearest Neighbor Model**

### Support Vector Machines

The Support vector machine model is known to be memory efficient and effective in high-dimensional spaces. However, we decided not to use this particular model because of its complexity, we researched that SVMs can be complex to tune as well as its interpretability, wherein understanding the decision boundaries for this model is more advanced and challenging than using logistic regression.



**Figure 3.4 Support Vector Machine Model**

#### **D. Oversampling Technique**

- We also tried the data handling technique synthetic minority oversampling technique (SMOTE) because of the imbalance in features. The recall, F1, and AUC significantly improved; however, the precision also decreased by approximately 20%. We decided not to sacrifice the precision of the model and retain the result obtained from the SMOTE because we believe that high precision is important for identifying whether a customer will purchase the year-end promo. It is crucial to have a high fraction or percentage of retrieved instances that are relevant (precision), not just the fraction or percentage of relevant instances that are retrieved (recall), as well as the harmonic mean of precision and recall (F1).

## **IV. RESULTS AND ANALYSIS**

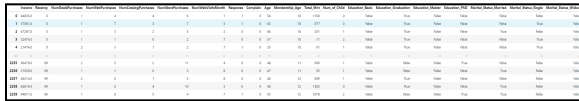
This section will contain the results obtained from the Logistic Regression Model created by the group. Discussion about the obtained results will also be incorporated here.

### **A. Outliers Removal**

As mentioned in the Methodology section of this paper, outliers were detected in the Income column of the given dataset. We employed Z-scores to detect and remove these outliers. This process involves calculating the Z-Score for the Income column itself, with a threshold set to 3 for this experiment. This threshold determines whether a record has an outlier income. After implementing this approach, we identified 8 records as outliers and removed them from the dataset. Prior to the removal of outliers, the dataset had a shape of 2240 rows x 22 columns, while after the removal, it was reduced to 2232 rows x 22 columns. Figure 2.6 displays the box plot of the Income feature after outlier removal, clearly illustrating the absence of outliers.

## B. One-Hot Encoding

After cleaning and standardizing the data, which is discussed in the methodology part of this paper, we have employed the use of One-Hot Encoding to convert categorical data into a numerical one so that the machine learning model can interpret the said data. To be specific, we applied the said technique in the “Marital Status” and “Education” columns of the dataset. After applying the said approach, the total size of the dataset became 2232 rows  $\times$  20 columns. Figure 4.1 shows the said dataset.



**Figure 4.1:** Dataset After One-Hot Encoding

## C. Modeling

Similarly to our professor’s demonstration and other tutorials that we have watched, modeling the machine learning will start by separating the features into an X variable, and the target variable, which is the Responses, is added on a y variable. The ID was not included in the features

selected to be used by the model since this does not contribute anything to predicting the responses of the members. Before getting the results of the model, we used hyperparameter tuning to ensure that the results were as accurate as possible. The following section will discuss the processes of the said approach.

## D. Hyperparameters Tuning

Applying hyperparameter tuning is essential in our machine learning model since this drastically increases our prediction model outputs results. In the methodology part of this paper, we show the difference between when the said approach is applied and when it is not applied. The process of Hyperparameter Tuning is a method for systematically working through multiple combinations of parameter tunes, cross-validating each combination to determine the best one. The following are the used variables for this approach:

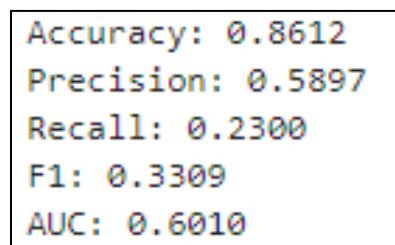
- **C** - this variable is a regularization parameter in the model we choose,

which is logistic regression. Based on our research, the lower the number indicated on this parameter, specifies stronger regularization. In our case, we have indicated [0.001, 0.01, 0.1, 1, 10, 100], which represents the range of regularization to be tested.

- **penalty** - this variable specifies the norm penetration used, and for this exercise, we used l1 and l2. l1 focuses on absolute value coefficients, while l2 focuses on squared value coefficients.
- **solver** - this variable specifies the algorithm to be used in the regularization optimization. We have used liblinear which is applicable most of the time for small datasets.
- **max\_iter** - this variable is used to specify the total number of iterations for the optimization algorithm to

converge. For this exercise, we used 100, 200, and 300.

These variables will initialize the GridSearchCV together with the logistic regression and the hyperparameter grid. This will also take the number of cross-validation (cv) and scoring. To output the best combination for the hyperparameters, the fit method is used on GridSeachCV with X and Y train. Figure 2.8 in the previous section shows the best hyperparameters obtained after applying the processes. These hyperparameters will be used to get the Accuracy, Precision, Recal, F1, and AUC. Figure 4.2 shows the result of the logistics regression prediction model.



```
Accuracy: 0.8612
Precision: 0.5897
Recall: 0.2300
F1: 0.3309
AUC: 0.6010
```

**Figure 4.2:** Model Results

Our group prioritized and focused on the model that performed better on the precision metric because precision measures the likelihood of the model to predict if the customer will purchase the year-end promo.

This is done by computing the correctly identified customers who will purchase the year-end promo (true positives) divided by all predicted customers to purchase the offer (sum of true positives and false positives). With the model centered around the precision metric, the predictions it outputs are assured to be trustworthy. Moreover, given that the dataset is imbalanced due to the numerous number of customers not purchasing the promo based on the response column, precision captures most of the customers who are likely to actually avail of the promo. This metric takes into account crucial factors in evaluating the model's performance to identify the minority class, which is the customers availing the promo.

## **V. CONCLUSIONS AND RECOMMENDATION**

We started this lab exercise 2 with a basic understanding of how machine learning models function. However, this time around, we encountered a notable difference: the availability of various models and a more extensive set of features, primarily numerical in comparison to lab 1. While lab exercise 2 was manageable, our challenge arose when attempting to increase precision. We noticed that when we managed to enhance precision, the recall decreased. Consequently, we explored alternative methods and approaches to enhance both metrics. However, our efforts resulted in an unexpected outcome because of the trade-off in the results: although precision improved, recall, F1 score, and AUC decreased. Subsequently, when we successfully increased recall, F1 score, and

AUC, we encountered a decline in accuracy and precision, leaving us at an impasse.

There are still improvements to be made in our experiment, particularly regarding the results produced by our predictive model. The results fall short of our expectations, which may be attributed to the features utilized in the model. We suggest incorporating additional features that could potentially lead to better outcomes. In addition, we also recommend exploring other machine-learning models. However, due to time constraints, we were unable to explore alternative machine learning models from the options available. Therefore, it would be beneficial to investigate other models suitable for addressing this type of problem. Furthermore, we also recommend exploring and researching more on the hyperparameters and how each affects the model to better understand the model's performance and prediction.

Overall, creating this prediction model has been a valuable experience as it has exposed us to learning about five new machine learning models.