Lab Exercise 2

CS ELEC 2C

Alessandro Andrei Araza Joshua Kyle Entrata

February 27, 2024

Abstract

ABC Supermarkets needs to do an analysis of whether or not their previous customers will accept their year-end sale of offering a gold membership for a cheaper price of $499 compared to $999 on the usual days. They wanted to know if their phone call campaigns would accepted by making a predictive model to classify which customers would purchase the offer. For implementation, logistic regression, support vector machines, naive bayes, decision trees, and k-nearest neighbors models were used to evaluate the dataset. Models had undergone cross-validation to ensure the consistency of reuslts. It was found throughout all of the models that someone's year of birth, their purchases (regardless of at the store, online, or what did they buy), and their marital status had a play at determining if they would accept the offer.

I INTRODUCTION

Supermarkets, like any other business, need to meticulously and carefully plan out their business decisions because one wrong move could result in millions in revenue losses. However, risky business decisions are what often create the most capital, popularity, and overall growth for a company, as these decisions create new avenues by attracting more customers or strengthening existing ones. Risky business decisions do not always need to be a gamble; there are many ways to turn them into a calculated risk. This approach allows a company to somewhat have an idea of the possible gains, losses, and approaches to the risks. The most traditional method for calculating risks is by listening to the veterans of the business. However, there are now more effective ways to help business decisions, one being leveraging data. Data and machine learning can help by providing models that predict outcomes to an accurate level that would help in the approach to risky business decisions.

1.1 Problem and Dataset

The lab exercise is provided a dataset that contains various pieces of personal information about customers. Specifically, the dataset contains the columns as shown in table 1.

The `Response` column will be the target for the machine learning models. To interpret, the goal of the machine learning models is to predict which customers accepted the offer in the last campaign. Doing so would give ABC Supermarkets an idea who might be willing to accept the offer of their year-end sale campaign for existing customers.

Table 1: Table of the dataset attributes and their data type

Column	Data type
ID	int64
Year_Birth	int64
Education	object
Marital_Status	object
Kidhome	int64
Teenhome	int64
Dt_Customer	object
Recency	int64
MntWines	int64
MntFruits	int64
MntMeatProducts	int64
MntFishProducts	int64
MntSweetProducts	int64
MntGoldProds	int64
NumDealsPurchases	int64
NumWebPurchases	int64
NumCatalogPurchases	int64
NumStorePurchases	int64
NumWebVisitsMonth	int64
Response	int64
Complain	int64

II METHODOLOGY

The standard workflow involves data analysis, preprocessing, modelling, evaluation. Apart from the `ID` column, all other data were fed into the model. There were a lot of preprocessing techniques used before the modelling, whichever were used and not used will be discussed in the

experimentation section of the paper. For now, below are some of the preprocessing methods done on the data: removing invalid values, removing null values, removing outliers, one hot encoding, interaction features, scaling, polynomial features

2.1 Modelling

Various classification models have been compared with each other. The models tested are Logistic Regression, Support Vector Machines, Naive Bayes, Decision Trees, and K-nearest neighbors

Modelling also involved cross-validation to ensure the consistency of the metrics.

III EXPERIMENTS

3.1 EDA

Exploroatory Data Analysis (EDA) provides an overview of what could be done to the data to improve the models' performance.

3.1.1 Data Profiling and Malformed Entries

All of the columns and their data types can be seen at table 1. When checking for null values, all other columns are free from null values other than `Income`. Upon inspection, there seems to be no connection with all the other attributes of the entries with null valued 'Income'. As far as inspection and analysis goes, they seem to just be malformed entries.

`Marital_Status` also has multiple entries that could be thought of as different, similar, or sometimes even a malformed entry. In the list of unique values for the column, the values are `Divorced, Single, Married, Together, Widow, YOLO, Alone, Absurd`. Upon listing the value counts, `Alone, YOLO, Absurd` have very few entries which could be considered as outliers. There have been thoughts of merging some of these entries to the other entries with bigger value counts; e.g. `Single` would adopt the records with `Alone`, etc. However, the problem lies wherein there is no way to deduce their backgrounds and whether or not the adopting categories should take in the outliers. To further explain, there is no way to tell if a record that is `Alone` might have came from a divorced background or if they simply are single. However, a further explanation of the choice between data integrity and model performance will be provided later in experimentations.

3.1.2 Univariate Analysis

The skewness and imbalance of data attributes become apparent upon further isolated inspection. Firstly, `Income`'s central tendencies at table 2

To put into perspective figure 1 shows the distance of the max value from the mean of the distribution with a skewness of 6.763487372811116

The decision to remove outliers depends on the models' robustness to outliers, which will be further explained in the experimentation section.

Table 2: Income column's descriptive statistics

Statistic	Value
count	2216
mean	52247.251354
std	25173.076661
min	1730
25%	35303
50%	51381.5
75%	68522
max	666666

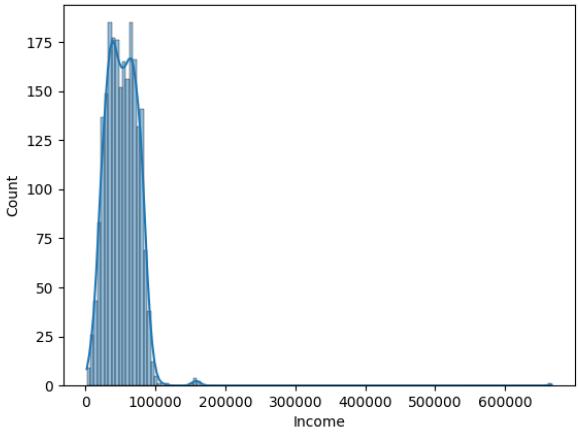

Figure 1: histogram of Income values

For `Education`, although `Basic` records could be considered as outliers, removing them may misrepresent the dataset.

3.1.3 Bivariate Analysis

The relationships of attributes to each other are also worth considering to look at. Given that the analysis of these attributes will later be interpreted by the models, this will be discussed in the later sections.

Looking at the scatterplot for the income per year of birth, ignoring the outliers, it seems that there seems to be an equal distribution for each year. Looking around the years 1970 to 1980, there seems to be a few entries where the recorded income is higher than the usual.

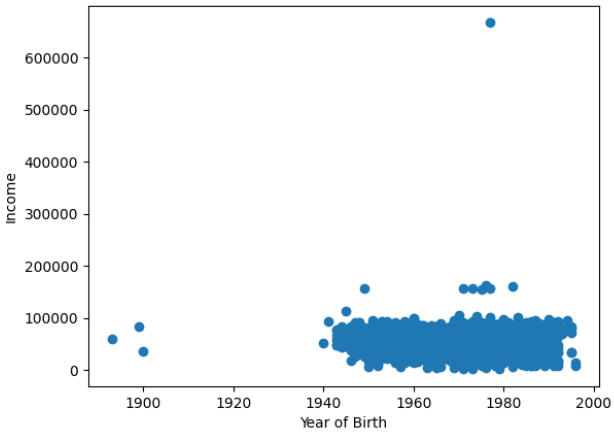

Figure 2: Scatterplot of income per year of birth

Regarding education, when categorized and calculated for central tendencies, `Basic` provides the lowest average income, while a `PhD` provides the highest average. However, according to table 3, Basic education seems to be the most stable when it comes to delivering a salary based on the standard deviation. PhD education, although high in the average salary, is also very high in standard deviation. This could be seen at the *min* and *max* of each category where the minimum income of someone with PhD degree can be as low as 4023 while someone with basic education can be sure to have a salary that is atleast 7500.

Table 3: Descriptive statistics for each unique value in the Education column

Education	$\bar{x}$	σ	min	max
2n Cycle	47633.19	22119.08	7500	96547
Basic	20306.26	6235.07	7500	34445
Graduation	52720.37	28177.19	1730	666666
Master	52917.53	20157.79	6560	157733
PhD	56145.31	20612.98	4023	162397

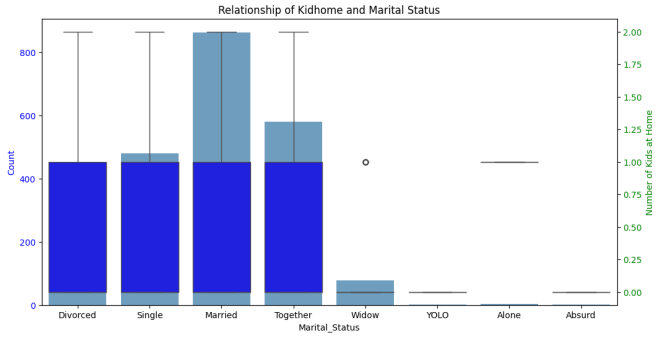

Figure 3: Relationship of Kidhome and Marital Status

The graph illustrates that `Divorced`, `Single`, `Married`, and `Together` statuses have higher counts compared to `Widow`, `YOLO`, `Alone`, and `Absurd`, which have fewer individuals. This visualization can help to give meaninful insights on the count and average of the number of children present in the customers' household. For instance, `Alone` has the highest average among the marital statuses but has a very low count. Displaying their average only can be misleading, which must be avoided as much as possible.

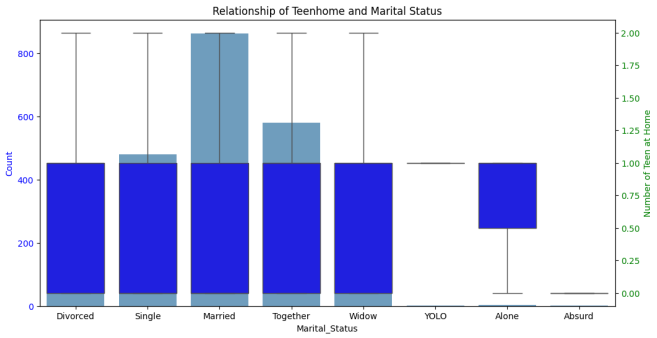

Figure 4: Relationship of Teenhome and Marital Status

Similar with the graph of the relationship of kids at home and marital status, the graph of `Teenhome` and `Marital_Status` resulted with a greater number of individuals with marital statuses of `Divorced`, `Single`, `Married`, and `Together`. In contrast, `Widow`, `YOLO`, and `Absurd` categories have significantly fewer individuals. Also, the `Married` category appears to have the highest average number of teenagers at home. This chart help in understanding the influence of marital status on the number of teenagers, which can provide valuable insights during feature engineering in the preprocessing phase.

3.1.4 Correlation Analysis

Using correlation analysis during the exploratory data analysis phase can be help identify relationships between variables and inspire the creation of new features. Understanding the impact of each feature to each other can help in building more efficient predicitve models.

3.1.5.1 Correlation between Income and Education. An investigation was conducted to analyze the relationship between `Income` and various `Education` levels. The `Education` was converted into distinct binary variables through one-hot encoding, so it can be used to quantify the correlation between each educational levels and `Income`. Individuals with a `Basic` education level exhibit a negative correlation (-0.200576), suggesting that this group tends to have a lower income levels compared to the baseline category. Meanwhile, the `PhD` category showed a positve correlation (0.081552), indicating that these individuals tend to earn more than the baseline, implying that the higher educational attainment can correlate with higher income levels. Lastly, the `2n Cycle`,

Graduation, and Master education levels showed correlations of -0.057745, 0.018935, and 0.011827, respectively, with Income. These figures suggest that they only have a slightly correlation with Income values.

These results exhibit a weak correlation between these two variables. This observation suggests that education alone is not a strong predictor of income levels within the dataset. This can become a basis to decide in which approach is the most appropriate during the preprocessing phase.

3.1.5.2 Correlation between Income and Marital Status. Similar to the examination of education's influence on income, correlation analysis between Marital_Status and Income was explored to provide meaningful insights that can be later used in the experiments. The Marital_Status was also converted into distinct categories through one-hot encoding for analysis with Income which yielded informative insights. Marital statuses Absurd (0.024026), Divorced (0.007975), and Together (0.0234425) both show slight positive correlations with Income, which suggests that individuals with these marital statuses tend to have marginally higher incomes compared to the baseline category. While, marital status Widow (0.031706) exhibit positive correlation as well, but has weak relationship. This status presents the highest positive correlation among the statuses. Lastly, marital statuses Alone (-0.012374), Married (-0.016479), Single (-0.025843), and YOLO (-0.004556) are negatively correlated with 'Income'. These finding imply that individuals with these marital statuses tend to have slightly lower incomes compared to the baseline.

Similar to the finding with Education, the correlations between Income and Marital_Status are generally weak. This indicates that Marital_Status is not a strong predictor of income within the dataset.

3.1.5 Chi-square test of independence

Using Chi-square test of independence is suitable in comparing categorical data to see if there is a statistically significant relationship between the two categorical variables.

3.1.6.1 Association between Income and Education. Customer complaints Complain and education levels Education was investigated to see their association with each other using the Chi-square test of independence. It resulted with a p-value of 0.11620258344593623, which is greater than 0.05. This suggests that there is no significant between the two variables, implying that Education is not a good determinant for customer complaints.

3.1.6.2 Association between Income and Marital Status. Similar with the observation between customer complaints and education levels, the association between Complain and Marital_Status was determined through Chi-square test. The test resulted in a p-value of 0.9870342644720567, indicating that there is no significant association between the two variables. Just like education level, marital status does not significantly affect whether customers have lodged complaints.

3.1.6 ANOVA Test

ANOVA tests were conducted to evaluate the relationship between Marital_Status and the variables Kidhome and Teenhome. These tests are important to understand how the number of kids and teenagers at home may vary across different marital statuses, which could potentially influence the development of the models and inspire to create new features.

3.1.7.1 Analysis between Kidhome and Marital Status. The ANOVA test for Kidhome by Marital_Status resulted to a F-statistic of 2.8150772422212915 and a p-value of 0.006399583359671683. A high F-value suggests that there are significant differences between the groups, while the p-value is below the significance level of 0.05. This suggests that there is a significant differences in the average number of children at home among different marital status. This means that the Marital_Status has a huge effect on the Kidhome variable.

3.1.7.2 Analysis between Teenhome and Marital Status. The ANOVA test for Teenhome by Marital_Status resulted to a F-statistic of 4.461449687945867 and a p-value of approximately 0.00006090978033406822 5. This p-value indicates that the number of teenagers in the household significantly varies by marital status. Given this significant p-value, it is safe to say that Marital_Status plays a bigger role in determining the Teenhome variable compared to Kidhome.

3.1.7 Analysis for Mnt[.]+ and Num[.]+ Columns

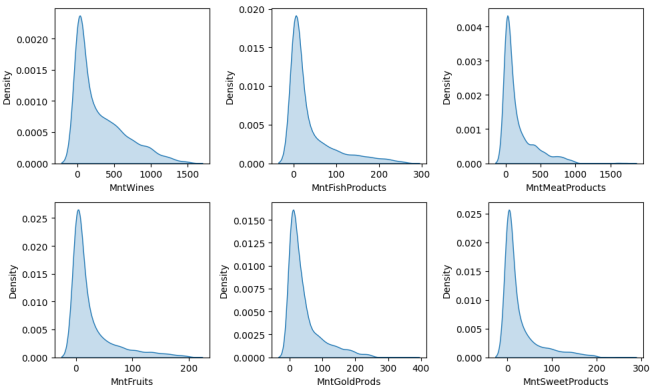

Figure 5: Distribution of Customer Spending Across Various Product Categories

The illustration consists of six kernel density estimation (KDE) plots, each representing the distribution of spednging across different products, such as wines, fish products, meat products, fruits, gold products, and sweet

products. Using KDE plots for these variables are helpful to visualize their distribution shape to identify the skewness of each spending category. All distributions are right-skewed, indicating that the majority of customers spend less in these categories. There is also a noticeable spending habits based on this graphs. For instance, `MntWines` show a potential for higher customer spending than other categories.

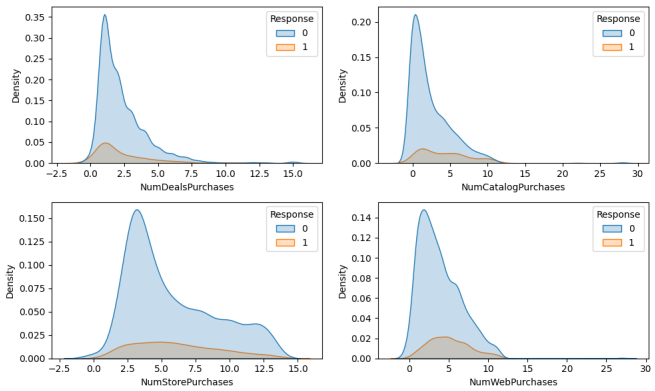

Figure 6: Differences in Customer's Purchasing Behavior Across Different Channels'

Similar with the illustration for the distribution of customer spending, KDE plots were used to illustrate the distribution of different types of purchases made by customers. Additionally, the customers' response to the campaign were segmented to see the pattern of their purchasing behaviors who responded to the campagaign and who did not. In general, those who availed the promo exhibit flatter distibutions in their purchasing behaviors across different channels which means that they are more likely to explore and buy using different channels. While the customers who did not avail the promo tend to buy less and not venture into different types of purchases than those who did.

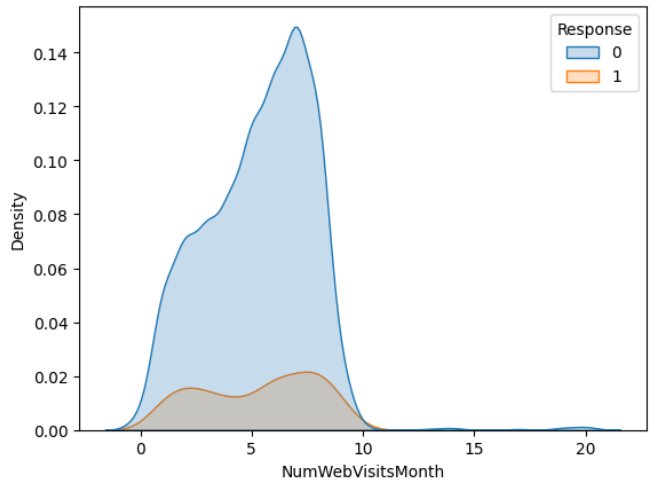

Figure 7: Density Plot of Monthly Website Visits by Respone'

The graph suggests that customers who did not avail the promo actually visit the website more frequently than those who got the promo. This could mean that those who did not avail are looking for deals in the website but failed to find what they want or those who got the promo are already satisfied customers that don't need to visit the website as often. This can be helpful in understanding the behavior of the customers and create meaningful strategies to create an accurate predictive model.

3.2 Preprocessing

3.2.1 Data Cleaning

3.2.1.1 Handling null values. During the preprocessing phase, one of the important steps was to address the null values, particularly within the `Income` column. Several solutions were considered for dealing with the null values in this column.

First, it was proposed to fill null values with the median of their respective groups. This method imputes missing values with the median income of similar groups based on other features, such as `Education` or `Marital_Status`. But after examining the relationship between `Income` and these categorical variables, we observed weak correlations: - For `Education` levels, the correlation coefficients ranged from -0.200576 to 0.081552 which indicate a weak relationship between these features. - Similarly, for `Marital_Status`, the correlation coefficients ranged from -0.025843 to 0.0031706, also suggesting a weak relationship between these features.

Given these weak correlations, imputing null values based on the meadian of their respective groups might not provide a meaningful impact on the models. It may also introduce inaccuracies into the dataset because neither `Education` nor `Marital Status` strongly predict `Income`. After testing this approach, the F1 scores of Logistic Regression, SVM, Decision Trees, and K-Nearest Neighbors significantly decreased with 0.045786, 0.04433, 0.149474, and 0.099193, respectively, less in scores.

Second, another approach for these missing `Income` values was to fill it with 0. This method can be justified if a null value represents customers' lack of income. However, this assumption is generally not accurate for most of the datasets. It could significantly skew the data distribution and inflate the count of customers' with zero income. Therefore, filling null values with zero was viewed as not optimal due to its potential to misrepresent the information of the customers in the dataset. The only model that decreased in F1 score using this approach is the Decision Tree which resulted in 0.036149 less in score.

Lastly, given the drawbacks of the first two approaches, it was decided to remove rows with null values. This decision was made by considering the analysis done in the features and preserving the integrity of the data. Although this approach may result in a smaller dataset, it ensures that reliability and complete information, which

reduces the potential for skewed results and maintain data quality.

3.2.1.2 Handling Outliers. Outliers were observed within the `Income`, `Mnt.*`, and `Num.*` columns during the exploratory data analysis phase. Visualization tools, such as kernel density estimate and scatter plots, indicated the presence of extreme values that could potentially affect the modelling. The following algorithm for removing outliers on continuous variables was implemented:

```
remove_outliers(data, column, z_thresh=2):
    z = absolute_value(zscore(data))
    outlier_indices = array_where(z > z_thresh)
    no_outliers = data.drop(outlier_indices)
    return no_outliers
```

Figure 8: Removing Outliers

Applying this algorithms ensures that the analysis is not skewed by extreme values, leading to more accurate results and improves the quality of the models.

3.2.2 Feature Engineering

3.2.2.1 Creating Total Children Feature. A new column called `Total_Children` was created by summing up the `Kidhome` and `Teenhome` columns. This new feature provides another information for the total number of children in each customer's household, which could help in understanding the purchasing behavior or creating new marketing strategies based on this new feature.

3.2.2.2 Removing Unrealistic Customers based on Year of Birth. During the exploratory data analysis phase, a number of outliers in the `Year_Birth` column was recognized, that is why the dataset was filtered to only include records where the column value is 1940 or later. This ensures a realistic age range for customers.

3.2.2.3 Converting dates to Datetime and validation. The values in the `Dt_Customer` column was converted to a datetime type. To ensure the dataset contains valid dates, checks were implemented to verify that there are no future dates and that the year of birth is not later than the year of becoming a customer.

3.2.2.4 Calculating Days Since Becoming a Customer. The number of days since the date of becoming a customer up to the current date was calculated. This new feature, `Days_Since_Customer`, was created to help in understanding the customer tenure, which can be important in models to predict customer loyalty.

3.2.2.5 Cleaning Marital Status Values. There are two approaches proposed in cleaning these values. At first, dropping the rows with `Marital_Status` values of `YOLO`, `Absurd`, and `Alone` was considered. However, this approach may possibly negatively affect the integrity of the data. It also resulted to a lesser F1 scores in SVM and Decision Tree models with a result of 0.027663 and 0.027541, respectively, less in scores.

That is why replacing this status labels with `Single` was done to maintain data size and integrity. This homogenization helps in reducing the variability within the data. This helps in decreasing the number of different values which can help in making the analysis more straightforward.

3.2.3 One-Hot Encoding

Columns `Education` and `Marital_Status` are categorical values. While there is an option to use factorization for these data, there is a risk of misinterpretation since the factorized values could imply that one number has a greater value than the other, which could lead to misinterpreation during the modelling process.

3.2.4 Interaction Features

Given the statistical evidence from the ANOVA test - $F = 2.815$ for `Kidhome` and $F = 4.461$ for `Teenhome`, with p-values well below the 0.05 significance level - during the explroatory data analysis phase, the results shows strong evidence that `Marital_Status` significantly impacts the number of children and teenagers in a household. This suggests that marital status may interact with household composition that may provide a meaningful impact on the model. To quanitify these relationship, interaction features were created by multiplying each one-hot encoded marital status column by `Kidhome` and `Teenhome` to create new interaction columns. These interaction features can potentially imrpove the performance of the models. They allow models to provide deeper insights on how households interact with the customers' marital status.

There were discontinued approaches using interaction feature between `Total_Children` and one-hot encoded `Marital_Status` because during the exploratory data analysis phase,

3.2.5 Feature Selection

The objective was to construct a predictive model to classify customers who might purchase the new year-end sale offer based on variables that may have a direct or indirect impact. Through careful analysis, there are columns that were dropped, such as `ID`, `Dt_Customer`, `Kidhome`, and `Teenhome`.

The `ID` contains unique identifiers for each record which are typically irrelevant to the outcome of predictive models. The date when the customer joined has been used to derive new feature, `Days_Since_Customer`, therefore the original date is no longer necessary for the modelling. The columns `Kidhome` and `Teenhome` were used to create interaction features and new feature, `Total_Children`. Retaining these original columns might be redundant and could even introduce noise.

Additionally, interaction features were introduced by using one-hot encoded marital status and variables like

`Kidhome` and `Teenhome`. Removing the original one-hot encoded columns can help in avoiding multicollinearity and potentially improve the performance of the models due to reduced dimensionality and complexity.

3.2.6 Scaling

Large distance between features can negatively affect the model's performance. For example, the `Income` value are in thousands, while the other variables, especially those that have undergone one-hot encoding only range in single digits. With the use of feature scaling, this would help the models in dealing with values that are too distant in range.

The use of `StandardScaler` helps prepare the dataset for many machine learning algorithms, particularly those sensistive to the scale of features, such as Support Vector Machines (SVM), and K-Nearest Neighbors (KNN).

3.2.7 Polynomial Features

While adding more features to a model doesn't equate to better performance, there is still a significant effect to a machine learning model when utilized correctly. The introduction of polynomial features would create a polynomial based on a single feature where each monomial would become an input in the model. This is based on assumption that the relationship between some variables could exhibit boosting when combined. By integrating polynomial feature, it can help the models with a comprehensive framework capture more complex relationships within the data. The addition of polynomial features notably improved the F1 scores for `Logistic Regression` (0.0717 increase), `SVM` (0.2065 increase), and `Naive Bayes` (0.0942 increase), showing their capability to capture non-linear relationships, while changes were little for `Decision Tree` (0.0011 increase) and `K-Nearest Neighbors` (0.0293 increase).

3.3 Sampling

To select which models would perform the best with a specific set of hyperparameters along with ensuring a level of performance that is not just attributed to the luck of a chosen `random_state`, there needs to be a repeated testing of the models' performance using different sets of data along with different sets of hyperparameters that modify how the models learn. For this, there is one main approach in dealing with consistency of performance and arbitrariness of choice: cross-validation.

3.3.1 Train-test-split and Cross-validation

Before cross-validation, a standard Train-Test-Split needs to prepend the sampling process. Consider the following example of simple k-fold cross-validation with 5 splits. The indices of the entries are within the range $[0, 100)$. table 4 shows the range of indices that would be within each fold:

Table 4: A simple example of a k-fold cross validation with a split/fold of 5 where the train and test columns show the index range of each set

Fold	Test Set	Train Set
1	[0, 20)	[0, 100) - [0, 20)
2	[20, 40)	[0, 100) - [20, 40)
3	[40, 60)	[0, 100) - [40, 60)
4	[60, 80)	[0, 100) - [60, 80)
5	[80, 100)	[0, 100) - [80, 100)

While there are training and testing samples, they simply go on rotation for each fold. This would in turn feed the model all of the samples and train with all of them, this is something to take caution especially for grid search algorithms. For a grid search algorithm to be able to determine the best hyperparameters, it would need to perform a cross validation algorithm with each possible combination of hyperparameters. If the model would be trained and evaluated on all of the dataset, there is a sort of "leak" due to the hyperparameters actually being catered to what's best even for the testing data. For testing or predictions to be valid, it must be unbiased, and the best way to make sure of this is to make sure that the models never see the testing set that will be used to evaluate the models' performance.

Hence, the traditional `train_test_split` algorithm was implemented. The algorithm would then train and evaluate itself by splitting the training of the dataset into smaller subsets of training and evaluating data. Refer to figure 10 for a diagram illustrating the split, including the traditional `train_test_split` before the cross-validation (figure taken from scikit-learn page).

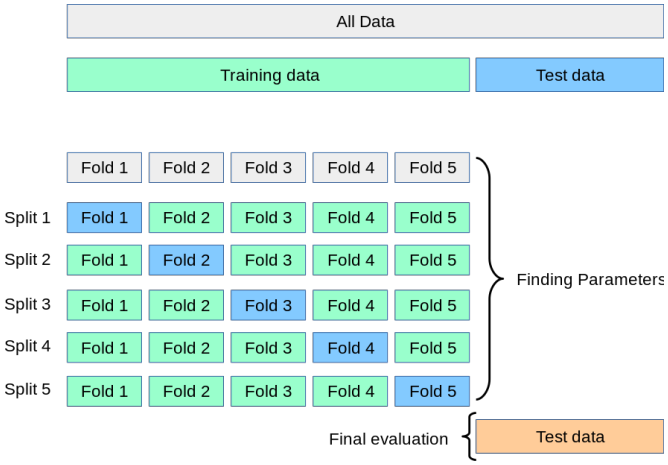

Figure 9: Illustration of the cross-validation folds after the initial train-test-split

3.3.2 GridSearch

The `GridSearchCV` algorithm takes an input of hyperparameters and a cross-validation method. The algorithm would try all combinations of hyperparameters and test them with the cross-validation method to assess their performance.

3.4 Model Selection

Using the `GridSearchCV`, it selects which set of hyperparameters works best on a model by training it with a cross-validation technique, but as mentioned earlier, only on the training data set. Note that for all of the grid search runs of each model, an f1 scoring system was used as the metric for measuring performance

3.4.1 Logistic Regression

The Logistic Regression is the simplest model out of all of them as a classifier, table 5 shows the performance for each combination of hyperparameters. The hyperparameters are just different numbers of iterations done for the solvers to converge.

Table 5: Logistic Regression cross-validation mean test scores for each hyperparameter

param_max_iter	mean_test_score	rank_test_score
50	0.399988	8
70	0.400451	7
100	0.413567	1
200	0.413567	1
500	0.413567	1
1000	0.413567	1
2000	0.413567	1
5000	0.413567	1

The learning performance of the model peaks and remains stagnant after that. It can be observed that even if the iterations are already as far as 5000 iterations, the scores remain the same. The learning curve of this model peaks around 100.

3.4.2 SVM

Table 6 shows the performance of the C-Support Vector Classification models. Note that for linear kernels, the gamma hyperparameter would not need to come into play. But the results show that the best performing model is the one with a linear kernel.

Table 6: C-Support Vector Classification models cross-validation mean test scores for each hyperparameter subset

gamma	kernel	mean_test	rank_test
	linear	0.393174	1
scale	poly	0.000000	4
scale	rbf	0.000000	4
scale	sigmoid	0.000000	4
	linear	0.393174	1
auto	poly	0.080769	3
auto	rbf	0.000000	4
auto	sigmoid	0.000000	4

3.4.3 Naive Bayes

A Bernoulli Naive Bayes Classifier, was used as a model for the dataset, table 7 shows the difference for each set of hyperparameter. An alpha of 7 with a fit prior of `True` seem to be the best model so far. However, when looking at the data for tests with the same value in `alpha` and `fit_prior` that is true, they tend to score higher with their counterparts with the same value in `alpha` but with a `fit_prior` of false.

Table 7: Bernoulli cross-validation test scores for each hyperparameter subset

alpha	fit_prior	mean_test	rank_test
1	True	0.321885	7
1	False	0.333608	4
2	True	0.326748	5
2	False	0.323729	6
5	True	0.337983	3
5	False	0.318559	9
7	True	0.345290	1
7	False	0.305031	10
10	True	0.339318	2
10	False	0.319661	8

There is another alternative to Bernoulli Naive Bayes which is the Gaussian Naive Bayes Classifier. However the distance between the precision and recall score are too far apart (refer to table 8).

Table 8: Comparison of Bernoulli NB with Gaussian NB

	Accuracy	Precision	ROC AUC	Recall
Bernoulli	0.345290	0.248854	0.787395	0.577778
Gaussian	0.171189	0.096033	0.536592	0.788889

3.4.4 Decision Tree

For Decision Trees, table 9 ranks that a log loss criterion with a standard min samples split of 2 all the while choosing the best split gives us the most optimal version of a Decision Tree Classifier. However, the the Decision Tree model that uses the entropy criterion isn't too far behind.

Table 9: Decision tree cross-validation test scores for each hyperparameter subset

criterion	min samples split	splitter	mean_test	rank_test
gini	2	best	0.371697	11
gini	2	random	0.346724	14
gini	5	best	0.309233	18
gini	5	random	0.373470	9
gini	10	best	0.329160	15
gini	10	random	0.378374	8
entropy	2	best	0.385656	7
entropy	2	random	0.372491	10
entropy	5	best	0.415986	2
entropy	5	random	0.317867	17
entropy	10	best	0.391090	4
entropy	10	random	0.361705	13
log_loss	2	best	0.416715	1
log_loss	2	random	0.388316	6
log_loss	5	best	0.390513	5
log_loss	5	random	0.365806	12
log_loss	10	best	0.399050	3
log_loss	10	random	0.324862	16

3.4.5 K-Nearest Neighbor

The hyperparameters adjusted were too many to be shown in a table. To be succinct, table 10 only shows the top 5 from the trials. The results show that regardless of the algorithm, whether it was a brute force search, a KDTree, or BallTree, or even if the model decides for itself which to use, The results remain the same so long as the number of neighbors is just 2 and the distance weight function is implemented.

Table 10: K-Nearest Neighbors cross-validation top 5 mean test scores

algorithm	n_neighbors	weights	mean_test	rank_test
auto	2	distance	0.297903	1
brute	2	distance	0.297903	1
ball_tree	2	distance	0.297903	1
kd_tree	2	distance	0.297903	1
auto	1	uniform	0.296767	5

Overall the performance of the models are summarized by table 11.

Table 11: Summary table of the scores of the best models chosen by grid search

	Accuracy	Precision	Recall	F1	ROC AUC
LR	0.899	0.475	0.389	0.414	0.771
SVM	0.875	0.365	0.433	0.393	0.752
NB	0.794	0.249	0.589	0.348	0.802
DT	0.885	0.410	0.400	0.431	0.676
KNN	0.884	0.348	0.267	0.298	0.629

IV RESULTS AND DISCUSSION

Due to the nature of the experimentation, there are five versions each implemented machine learning model, each with different hyperparameters. However, for the interpretation and discussion of results, only the best performing model are taken into consideration.

4.1 Logistic Regression results

Due to the polynomial features, there were 465 columns fed into the models, because of this, it's best to instead take note of the most significant coefficients from the model. Looking at table 12, it can be observed that the most significant features are all interaction features.

Table 12: Top 5 Features in the logistic regression model

Feature	Value
MntFruits A_Marital_Status_Single_Kidhome	0.979750
MntMeatProducts A_Marital_Status_Married_Teenhome	0.952025
MntWines A_Marital_Status_Together_Teenhome	0.844306
Education_Master A_Marital_Status_Single_Kidhome	0.814164
MntWines NumWebPurchases	0.763514

It is observed that people who bought fruits and are married with a kid at home are very likely to respond and accept the offer. The likeliness to accept the offer can also be seen with married people with a teen at home who also bought meat products in the last two years, people that live with someone (`Together` marital status value) that buys wines with a teen at home, single people who achieved masters education with a kid at home, as well as people who bought wines in the last two years that also bought products at the company's website.

4.2 SVM

Since the best model tested in the SVC models is a linear model, it can be interpreted that the coefficients to understand how the model has learned. Table 13 shows a lot of similarity with the top five coefficients of the linear regression model.

Table 13: Top 5 features in the SVM model

Feature	Value
MntWines NumWebPurchases	0.904773
A_Marital_Status_Married_Kidhome	
A_Marital_Status_Married_Teenhome	0.804857
MntFruits	
A_Marital_Status_Single_Kidhome	0.803837
NumWebVisitsMonth	
A_Marital_Status_Single_Kidhome	0.769760
Recency	
Days_Since_Customer	0.733147

As it turns out, the support vector machine model's top five coefficients has two features similar to the top five of the linear regression model. This would make sense because although they are different machine learning models, grid search algorithm has decided that the best hyperparameters for the SVM included that the svm should use a linear kernel. This would imply that the data is much akin to being linearly separable. Table 13 implies the same ideas as it did for the features that were also in table 12. For the features unique to table 13, they simply imply that people who are married with children and teenagers at home, single with a kid at home that visits the company's website in the past month, and people who haven't bought in a while (high `Recency` value) that are also long standing customers (high `Days_Since_Customer` value) are more willing to respond and accept the offer.

It is a bit difficult to interpret the support vectors found by the SVM. Mainly because any aggregation to these found support vectors are only slightly different to the aggregations found in the original dataset. For example, the average `Year_Birth` for the dataset is 1970.325371 while the average for the support vectors is 1970.030303; this is reflected to the other attributes even with different aggregation methods, whether it is the average, standard deviation, etc. the results only vary by a few units. This could be explained by the idea that maybe much of the data points are closer to decision boundary than expected. According to the model there are a total of 198 support vectors found in the dataset; 131 for no responses, and 67 for the responses. To put into perspective, out of the total 971 entries in the training set, 131 are support vectors, that is about 20.3% of the training set.

4.3 Naive Bayes

According to the implemented Bernoulli Naive Bayes, the log probability of a 0 response (not accepted) is -0.09726884, while a 1 response (accepted the offer) is -2.3785168. This means that the chance of accepting the offer is much lower in probability compared to rejecting it which would make sense given there is a heavy imbalance in the dataset where 1101 responses are 0 while only 113 entries have a response of 1.

When analyzing the log probability of a feature, the values imply the probability of a feature for a given class, i.e. $P(x_i \mid y)$. Just like the analysis of coefficients at the logistic regression and SVM models, there is too much features to be able to show them all. So, just as before, it is best to just look at the highest scoring (or alternatively, the lowest scoring) of the feature log probabilities.

Table 14: Top five highest feature probabilities for response 0

Feature	$LogP(x_i \mid y = 0)$
Year_Birth	-0.624380
Year_Birth Recency	-0.641402
NumCatalogPurchases	
Days_Since_Customer	-0.650023
NumCatalogPurchases	-0.650023
Recency	-0.650023

Table 15: Top five highest feature probabilities for response 1

	1
NumWebVisitsMonth	
Days_Since_Customer	-0.148846
NumCatalogPurchases	-0.239230
Year_Birth	
NumCatalogPurchases	-0.239230
NumCatalogPurchases	
Days_Since_Customer	-0.239230
NumCatalogPurchases	
NumWebVisitsMonth	-0.252835

Table 14 shows that someones year of birth and combined it with the recency of purchases, as well as the amount of catalog purchases with their length as a customer, it determines that they will likely not accept. Note however that these are continuous variables, hence these probabilities behave more like distributions. On the other hand, for people who did respond, their visits at the website combined with their days as a customer, as well as

their catalog purchases which can be combined with their year of birth days as a customer and visits at the website last month determine if they will accept the offer (see table 15)

4.4 Decision Trees

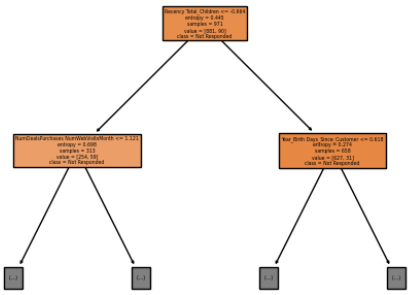

Figure 10: Illustration of Decision Tree

In the decision tree, the root node splits on the feature of `Recency Total_Children <= -0.664`. This indicates that the combination of these features is important in predicting the target variable, and uses the value of -0.664 as a threshold to give a two possible outcomes, considering that the features were standardized during preprocessing. If the condition is less than or equal to -0.664, then the left branch is taken; otherwise the right branch is taken. It implies that recency and number of children in the household are signifcanty factors in predicting customer response.

Moreover, the left child node, `NumDealsPurchases NumWebVisitsMonth <= 1.121`, and right child node, `Year_Birth Days_Since_Customer <= 0.818` further splits the data to represent more classes to predict the value of target variable. These child nodes highlight the importance of the number of deals and website visits, and the customer's age and days since they first became a customer in predicting their response.

V CONCLUSION

Overall the models have had similarities but at the same time differences when it comes to factoring in which inputs to give importance to and which do not have that much influence. Throughout the experimentation, linear regression and SVM models seem to lean more on classifying a customer based on their marital status, number of children at home, as well as what kind of product they buy at the supermarket. While for naive bayes, the the amount and frequency of purchases, combined with their birth year and time as a customer play much of a part than the others.

The decision tree's first decisions were to focus on income, purchases, and recency of purchases. But eventually, the final decision comes down to what type of product do they purchase, income, and your education.

All these models leverage data in very differing ways because each model is different, there is no one way to learn something and it shows in these models especially highlighted by the fact that they compute data differently. Or alternatively, perhaps the models are not as consistent as they should be because of their performances that could stand to be improved. But nonetheless, even at this current state, these models already coincide in some things.

It would be a good recommendation to improve training. It could be done by introducing more preprocessing techniques such as overampling, more feature engineering, etc. Additionally, features and preprocessing methods can be catered to each model.