WINE EXPENDITURE ESTIMATES

SEP 2023 // 510430519

Table of Content

Introduction

Business Context

As of 2023, the global landscape of digitalisation has undergone a rapid revolution, significantly propelled by the forces of globalisation and the COVID-19 pandemic, ushering the world towards a new digital paradigm and consequently reshaping consumer spending behaviour and market dynamics (McKinsey, 2020). In this increasingly digitised era, markets such as the food delivery platforms are becoming more competitive than ever on a global scale. Therefore, acquiring nuanced and accurately informed insights into consumer spending patterns on particular products is imperative for businesses like iFood, a food delivery company based in Brazil, to facilitate not only its dynamic efficiency in an era of escalating uncertainties but also empowers iFood to optimise their marketing strategies.

One of the intriguing factors of consumer spending within the domain is the money spent on wine products in the last two years for a given customer. Analysing, modelling and predicting the expenditure on wine for a given customer not only furthers the understanding about the purchasing power of the customer base who uses iFood, but also creating the personas of consumer who tends to spend more on wines, which can be pivotal for tailoring informed marketing strategies decision.

Problem Formulation

The primary objective of this project is to construct a predictive model that accurately forecasts a customer's expenditure on wine products over the last two years (response variable **`MntWines`**) as a function of various predictors. Three distinct models from various model families have been used to train, fine-tuned, validate, evaluate to predict the response variable **`MntWines`**, namely the linear regression (LR) model, KNN regression model and random forest regression model. These models were trained on a standardised sample of 1433 observations of customers from the training dataset, and subsequently validated on a standardised sample of 359 customers from the dataset where the model has not trained on. The model with the lowest MAE on the validation set will be selected as the final model, where the predictive performance of the model will be evaluated on the test set of 448 customers to assess the generalisability of the model. Despite the varying sample sizes of the training, validation and test sets, the number of potential predictors used in the models are kept constant, specifically 25 potential predictors excluding the variables that are not useful in predicting the response variable, which are then selected using different methodologies according to the model family.

Notably, the 25 potential predictors have covered a wide range of information regarding the customers, which can be classified into 3 categories: socio-demographic information, expenditure behaviour on different products and customer engagement with the company's marketing campaigns. Among the 25 potential predictors, the secondary objective of the project is to identify the most important predictor(s) that are useful in predicting the response variable, which will be discussed in the 'Model Selection' section. It is important to note that the final model and the variables selected are based on the goal of maximising predictive accuracy through minimising the MAE on the validation set, which measures the average

absolute difference between the actual and model predicted values of the expenditure on wine products over the last two years. Below shows the formula for MAE:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where n = sample size, y_i = actual amount spent on wine in the dataset, $\hat{y}_i$ = predicted amount spent on wine using the model

Despite MAE simplifies the evaluation of the model, it is important to note that it is sensitive to outliers, which may lead to a loss of information, which will be discussed in the 'Limitations' section.

Data Processing

Prior to Exploratory Data Analysis (EDA), it is essential to remove errors in the dataset, which includes missing values, collinearity between predictors, degenerate variables, and variables that are not useful in predicting the target variable by examining the training dataset. The reason for not examining the whole dataset is because it provides information on the test dataset, which would influence the model and variable selection process, leading to overfitting. It is also important to note that any transformation on the training dataset should be applied to the validation and the test set, ensuring the consistency, accuracy and interpretability of the model.

Encoding Categorical Variables

Given the goal of maximising the predictive accuracy of the model, it is essential to quantify the categorical variables (Appendix 1) into ones that can be analysed and visualised instead of dropping them, which ensures that no information is lost in the process of building the predictive model.

According to Appendix 1, variables **`Education`**, **`Marital_Status`** and **`Dt_Customer`** are of object type, indicating the categorical nature of these variables, which requires further encoding. As evident from Appendix 2, **`Education`**, **`Marital_Status`**, and **`Dt_customer`** have 5, 8, and 599 unique values respectively. Given the reasonable number of unique values in **`Education`** and **`Marital_Status`**, these variables have been encoded in discrete form (for example **`Education`** is divided into 5 categories, namely **`Basic`**, **`2n Cycle`**, **`Graduation`**, **`Master`**, and **`PhD`**. Each is assigned a number from 1 to 5). However, given the large number of unique values in **`Dt_Customer`**, it is not recommended to encode this variable using aforementioned encoding, as it will lead to a large number of columns in the data frame, which may lead to the curse of dimensionality. Given the **`Dt_Customer`** describes the date when the customer joined the company, it is more reasonable to create a derived variable, namely the age of the customer when they joined the company (**`age_joined`** = **`Dt_Customer`** - **`Year_Birth`**). As such, **`Dt_Customer`** and **`Year_Birth`** will be dropped from the data frame as **`age_joined`** is directly dependent on these two variables, but at the same time, increases the interpretability of the model (for example, if the age of the customer when they joined iFood is older by 1 year, we can expect on average an $x increase in the amount of money spent on Wine products in the last two years).

Degenerate and Useless variables in predicting the response

The decision to drop variables is based on two main criteria:

1. The variable is a degenerate variable, i.e. the variable has only one unique value. According to Appendix 2, degenerate variables are **`Z_CostContact`** and **`Z_Revenue`**. These variables will be dropped as they do not provide any information to the model.
2. The variable is useless or very limited in providing information to the response. From a parsimonious approach, it is strongly arguable that **`ID`** should be dropped as it is a naturally incrementing unique identifier for each customer, which presents no information regarding the customer's spending on wine products.

Missing values

By examining the training set, there is only one variable with missing values, which is **`Income`** (Appendix 3). Though the 15 missing values only account for 1.05% of the total training set, the decision is made to impute the missing values with the median of the variable rather than dropping the observation with the priority of maximising predictive accuracy. The choice of imputation is based on the distribution of **`Income`** as shown in Figure 1. The distribution of **`Income`** is right-skewed, with a long tail on the right. Therefore, the median is chosen as the imputation strategy as it is more robust to outliers than the mean.

To keep track of the imputed values, an indicator variable **`income_missing`** is created to indicate whether the value of **`Income`** is missing or not. The inclusion of this variable is to make sure the imputation strategy does not introduce any bias to the model, as the imputed values are not the true values of **`Income`**.

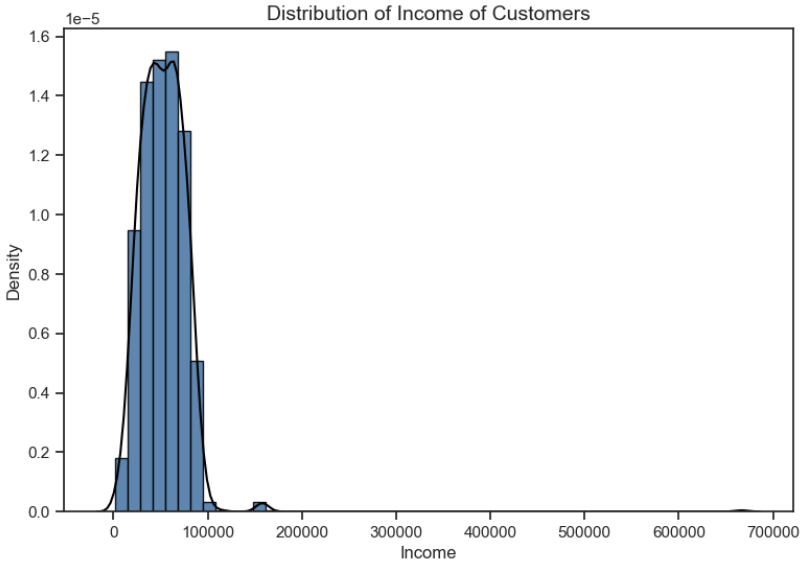


Figure 1: Distribution of income of customers in the training set

Perfect correlation between predictors

The final dimension to consider in terms of perfect collinearity between predictors (where correlation = 1), in which case one of the variables should be dropped to avoid collinearity.

Figure 2 below is a filtered correlation matrix, where only the variables with correlation = 1 are shown. Evidently, it shows that no two variables are perfectly correlated except the diagonals (where correlation with itself is expected to be 1). VIF is then performed, with the results of VIF values all below 5, validating that no predictors need to be dropped due to perfect collinearity.

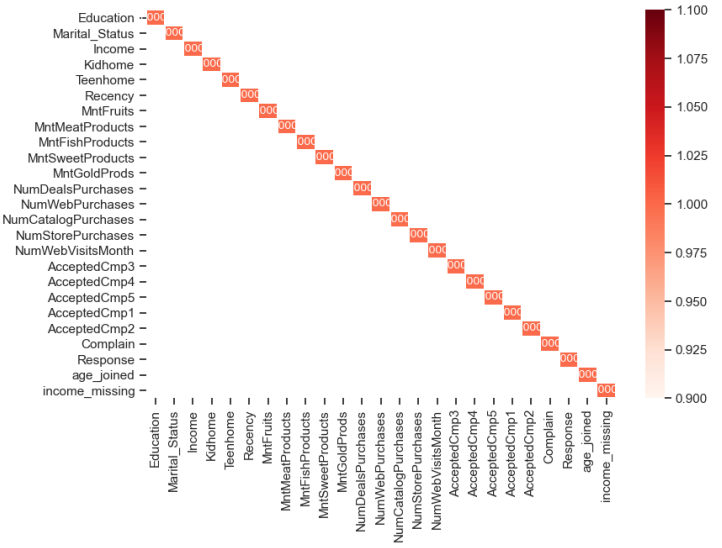


Figure 2: Correlation matrix that displays perfect correlation between predictors (corr = 1)

Standardisation

Lastly, given the different scales of the predictors, standardisation will be performed on the training set before modelling. For instance, **`Income`** is measured in dollars, while **`NumWebVisitsMonth`** is measured in number of visits. Standardisation will ensure that the predictors are on the same scale, and the model will not be biased towards predictors with larger values.

EDA

EDA is performed on the cleaned training dataset to add value in the understanding and familiarisation with the behaviour of the response variable, and bivariate relationships between the response variable and the explanatory variables. Such understanding would result in a more comprehensive interpretation of the model results.

Response variable

From Figure 3, **`MntWines`** has a large standard deviation of 334.5343 (4.d.p) relative to the mean of 305.4787 (4.d.p) which, coupled with the large range of 1492, suggests that the response variable is highly variable. Given this, the later section would discuss the size of the coefficients and prediction errors when discussing the model. Additionally, the amount spent on wine products is right-skewed (as indicated by a skewness of 1.1180), indicating that there are more customers in the past two years who spend less on wine products than those who spend more. Furthermore, the normal QQ plot (Figure 4) presents deviation from the normal distribution as represented by the red line. This suggests that the **`MntWines`** violates the normality assumption, which is a important assumption for LR, undermining the reliability of the model.

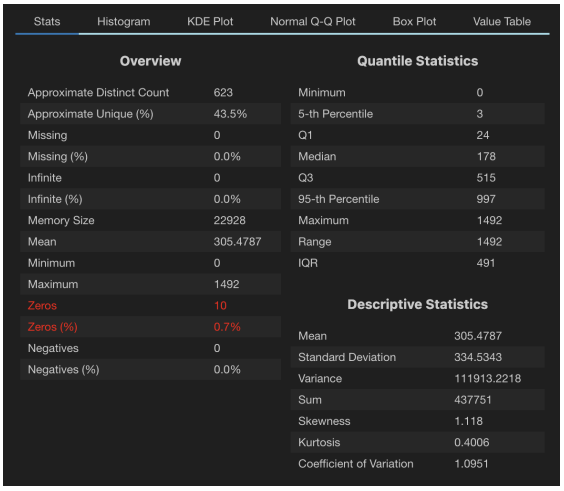


Figure 3:Descriptive statistics of the amount spent on wine

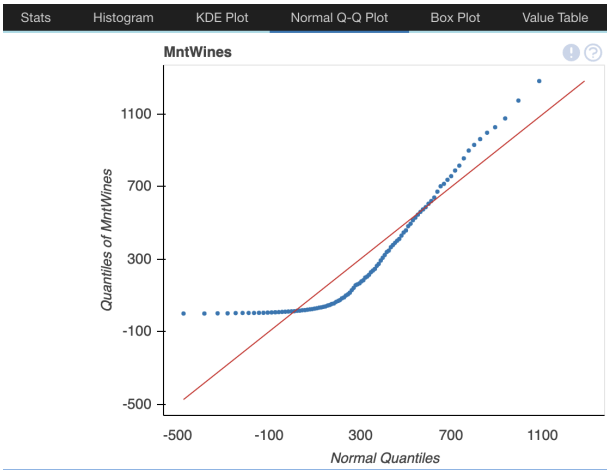


Figure 4: Normal Q-Q plot of the amount spent on wine

Bivariate Relationship

From the pairplot (Appendix 4), predictors **`Income`**, **`MntMeatProducts`**, **`NumWebPurchases`**, and **`NumCatalogPurchases`** presents certain level of positive linear relationship **`MntWines`** (Figure 5), which suggests that these predictors may be good predictors of **`MntWines`**. These predictors also intuitively reasonable to results in a higher amount spent on wine products. Given this, in the model selection process, we can compare the set with the sets of predictors selected by the models to add another layer of evaluation.

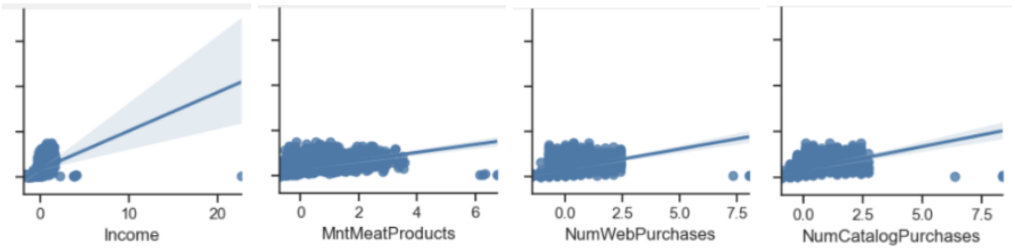


Figure 5: Extraction of the pairplot between response (`**MntWines**` on the y-axis) and the predictors (x axis)

Model Selection For Primary and Secondary Goal

The primary goal is to find a model that predicts accurately the amount spent on wine products in the last two years for a given customers, with the prediction error measured by the MAE. Among the models considered, the secondary goal is to find the main predictors that are significant in predicting the amount spent on wine products.1. Linear Model - Linear Regression

1. Linear Model

1.1 Variable Selection | Automatic Forward and Backward Selection

The first model is the linear model, specifically the LR model. In light of the large number of predictors, forward and backward selections were performed to select the best subset of

predictors that minimises the MAE. The mechanism behind the forward selection is that the model starts with a null model, and iteratively add on to the model the predictor that maximises the negative mean absolute error. The reason for maximisation seems counterintuitive, but it is because the score function is negating MAE. The expression below is the LR model that uses the predictors selected from the forward selection method (Appendix 5 contains all the coefficients of the selected variables):

$$\hat{MntWines} = 305.4787 - 5.2754 * Education + 38.2844 * Income + \ldots + 18.9215 * AcceptedCmp2$$

Since forward selection is a greedy algorithm, it is not guaranteed to find the best subset of predictors that minimises the MAE. Therefore, backward selection is also performed in the varaible selection process. The backward selection starts with a full model, and iteratively remove the predictor that maximises the negative MAE. The expression below is the LR model that uses the predictors selected from the backward stepwise selection method (Appendix 6 contains all the coefficients of the selected variables):

$$\hat{MntWines} = 305.4787 - 5.2754 * Education + 36.5375 * Income + \ldots + 18.9434 * AcceptedCmp2$$

Both implementation of the forward and backward selection are done using the mlxtend library, which is a more exhaustive algorithm that searches through all possible combinations of predictors to find the best subset of predictors that minimises the MAE.

1.2 Interpretation of the Model

The constant term (305.4787) refers to the estimated amount spent on wine products when all predictor variables are set to zero. However, this is not a meaningful interpretation as there are no zero values allowed in some of the predictors such as **`Education`** and **`Marital_Status`**.

The interpretation of the coefficients of the predictors must be interpreted with caution that the predictors has been standardised. For example, the coefficient of **`Income`** (38.2844 or 36.5375) suggests that for every one standard deviation increase in **`Income`**, the amount spent on wine products increases by $38.2844 or $36.5375 for forward and backward selection respectively, holding all other predictors constant. With negative coefficient like -21.0217 for **`MntSweetProducts`** in the forward LR model, it suggests that for every one standard deviation increase in **`MntSweetProducts`**, the amount spent on wine products decreases by $21.0217, holding all other predictors constant. This aligns with the natural assumption that customers who spend more on sweet products are likely to be accompanied by children, and hence they may spend less on wine products.

1.3 Evaluate Predictive Performance

The final linear model is selected based on the lowest MAE on the validation set as it suggests the model is more accurate in predicting the amount spent on wine products on average rather than overfitting to the training set. According to Table 1, LR using automatic forward selection is the best model under the linear modal family, with the lowest MAE of 113.2355, which has also demonstrated a good performance on the test set with a MAE of 120.1147, 0.4 lower than the LR using automatic backward selection. Figure in Appendix 7

provides a visualisation of the best LR model in predicting the amount spent on wine products on the test set.

	MAE on validation set	MAE on test set	main predictors (secondary goal)
Linear Regression using automatic forward selection	113.2355	120.1147	Education, Income, MntFishProducts, MntSweetPr...
Linear Regression using automatic backward selection	113.2443	120.5487	Education, Income, Teenhome, MntSweetProducts,...

Table 1 : LR model performance between forward and backward selection

2. KNN Model

With the knowledge that the normality assumption has been violated, LR may not be the best model to use. Therefore, we will consider using non-parametric models like KNN to predict the response variable as it is more lenient to the normality assumption. The mechanism behind KNN model is that it predicts the response of a new observation by finding the k-nearest neighbours of the new observation, and then taking the average of the response values of the k-nearest neighbours.

2.1 Hyperparameter Tuning

With the goal of maximising predictive accuracy, hyperparameter tuning has been performed to find the best k for the KNN model using 10-fold cross validation. In order to find the best k, KNeighborsRegressor is used to fit the training set with different values of k until the best number of neighbours is found to minimise the MAE of the validation set.

2.2 Variable Selection and Final Model

Variable selection has been the most computationally intensive part, where the mechanism behind the variable selection is similar to the forward stepwise selection in linear regression, however instead of adding the best predictor into the best subset, an exhaustive search is performed for knn to find the best subset of predictors that minimises the MAE of the validation set in an attempt to maximise the predictive accuracy of the model. However, it is important to note that due to the large number of potential predictors, the maximum number of neighbours has been set to 10, highlighting the trade-off between computational efficiency and predictive accuracy. Ultimately, the best knn model is when k = 4 using Eucledian distance as the distance metric since the dataset has been standardised. Given this, working towards the secondary goal, the best subset of predictors for the knn model is **`NumCatalogPurchases`**, **`NumWebPurchases`**, **`Income`**, **`NumWebVisitsMonth`**, **`AcceptedCmp5`**, **`NumStorePurchases`**, **`MntMeatProducts`**, **and `MntFishProducts`**.

2.3 Evaluate Performance

Notably, the KNN model performs better than the linear model in terms of having a lower MAE on both the validation and test set. Figure in Appendix 8 provides a visualisation of how the 4NN model in predicting the amount spent on wine products on the test set.

	MAE on validation set	MAE on test set	main predictors (secondary goal)
KNN: 4NN Model	91.1692	87.9202	NumCatalogPurchases, NumWebPurchases, Income, ...

Table 2 : KNN model performance with the best k

3. Random Forest Regression Model

Despite encompassing both parametric and non-parametric models, the limitations of LR and KNN models, specifically that LR relies on the linearity assumption between `MntWines` and the predictors in which case it is uncertain whether the true relationship is linear or not, and KNN can is computationally exhaustive which may not be the best choice given the priority of maximising predictive accuracy, especially when KNN is relying heavily on the choice of k that requires computational exhaustive hyperparameter tuning to find the best k. In light of this, a more flexible model, namely the random forest model, can be used in an attempt to mitigate the limitations of the previous two models, whilst maintaining the predictive accuracy of the model.

3.1 Hyperparameter Tuning and Random Forest Model

Similar to KNN model, random forest regression model also requires hyperparameter tuning to find the best combination of number of estimators and maximum depth of the tree. However, random forest model is considerably less computationally exhaustive, as it supports automatic features interaction that KNN does not. Specifically hyperparameter tuning for random forest is performed through the RandomizedSearchCV function, which randomly selects a combination of **`n_estimators`** and **`max_depth`** from the grid of hyperparameters, and evaluate the performance of the model using 10-fold cross validation. By fitting the training data into the randomised search cv object, the best hyperparameters are found to be **`n_estimators = 90`** and **`max_depth = 18`**. The final random forest model is obtained by fitting the training set into the the random forest regressor object with the best hyperparameter found. Figure 6 shows the feature importance obtained from the random forest model, where **'NumCatalogPurchases'** has shown to be the primary driving factor of **`MntWines`**.

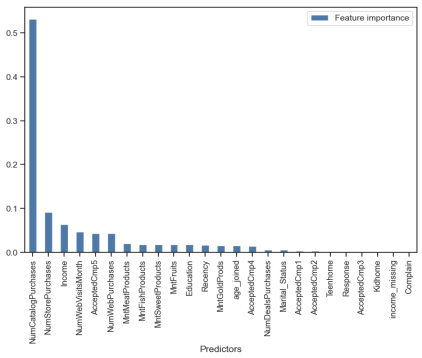


Figure 6: Feature importance based on the random forest model

3.2 Evaluate Performance

Figure 7 shows the predictive performance of the final random forest model on the validation and test set. Comparison between model performance is analysed in the 'Final Model Selection' section.

	MAE on validation set	MAE on test set	main predictors (secondary goal)
Random Forest Regression Model	93.7907	92.3611	[NumCatalogPurchases, NumStorePurchases, Incom...

Figure 7: Random forest model predictive performance

Final Model Selection

Figure 8 concatenates the predictive predictive performance of the three models in terms of MAE on both the validation and test set. Evidently, 4NN model has the lowest MAE on the validation and test set, which suggests that it is the best model among the three models in terms of accuracy. However, the trade-off is that the 4NN model is the most computationally expensive model both in terms of real time and CPU time. In the context of maximising predictive accuracy, the 4NN model is the best model to use. However, in the business context, the random forest model may be a better model as the predictive accuracy is not significantly worse than the 4NN model, but it is much more computationally efficient.

	MAE on validation set	MAE on test set	main predictors (secondary goal)
Linear Regression using automatic forward selection	113.2355	120.1147	Education, Income, MntFishProducts, MntSweetPr...
Linear Regression using automatic backward selection	113.2443	120.5487	Education, Income, Teenhome, MntSweetProducts,...
KNN: 4NN Model	91.1692	87.9202	NumCatalogPurchases, NumWebPurchases, Income, ...
Random Forest Regression Model	93.7907	92.3611	[NumCatalogPurchases, NumStorePurchases, Incom...

Figure 8: Models' predictive performance in terms of MAE

Additional Section

1. Best Model with One Predictors

The modelling process with one predictor is similar to the process with multiple predictors, except that the model is fitted with only one predictor. With linear regression model (SLR), the best predictor is obtained by comparing MAE of the model fitted with each predictor. OLS results (Appendix 8) shows that `age_joined` is the best predictor for the linear model, with the lowest MAE of 274.2544 and 263.1235 on the validation and test set respectively. For KNN model, hyperparameter tuning is no longer required as there is only one predictor (k=1) and distance is by default Euclidean. The best predictor for KNN model is `Income`, with the lowest MAE of 175.3231 and 165.4821 on the validation and test set respectively. Last but not least, the best predictor for random forest regression model is `NumCatalogPurchases`, with the lowest MAE of 93.7907 and 92.3611 on the validation and test set respectively.

Figure 9 shows the comparison of the MAE of the three models on the validation and test set, with random forest regression model having the lowest MAE on both sets, indicating that it is the best predictive model that uses a single predictor.

	MAE on validation set	MAE on test set	Best predictor
SLR	274.2544	263.1235	age_joined
1NN	175.3231	165.4821	Income
Random Forest Regression Model with 1 predictor	93.7907	92.3611	NumCatalogPurchases

Figure 9: Predictive performance of models with one predictor in terms of MAE

2. Asymmetric Error Adjustment

Given the predictive error is not symmetric, this means that predictive error metric MAE should be adjusted through penalisation parameter **`alpha`**, which has been set as 0.7. The choice of penalisation on underprediction and overprediction is determined by the distribution of the response variable (**`MntWines`**). From the EDA, it is evident that the response is right-skewed, indicating that more customers are spending less on wine than those who spend more. Therefore, given the prediction error is asymmetric, we will penalise the model more when it overpredicts the amount spent on wine products than when it underpredicts the amount spent on wine products. Therefore, the adjusted MAE metric is given by:

$$MAE_{asymmetric} = \frac{1}{n}\sum_{i=1}^{n} L(y_i - \hat{y}_i)$$

Where $L(y_i - \hat{y}_i) = \alpha \times |y_i - \hat{y}_i|$ if $(y_i - \hat{y}_i) < 0$(overprediction);

$L(y_i - \hat{y}_i) = (1 - \alpha) \times |y_i - \hat{y}_i|$ if $(y_i - \hat{y}_i) \geq 0$(underprediction)

Due to the extensive runtime issue of refitting the models with the adjusted MAE metrics accounting the asymmetric error, the subsequent comments on the results remain hypothetical. The hypothesized result is that the models will perform better in terms of predictive accuracy in the validation and test sets, as the models are penalised more heavily for underestimating the amount spent on wine products than overestimating the amount spent on wine products.

3. Fairness Analysis

To account for the socio-economic bias inherent in the dataset and thus the best model derived from the dataset, a fairness analysis is conducted on commonly percepted as vulnerable groups in society, specifically the protected groups identified in the project are widow with children and people who are below graduate level of education.

From the outset, the aforementioned protected groups is first identified in the dataset by binarising the socio-demographic categorical variables into two groups, namely the **`is_educated`** (0 indicates people who are below graduate level of education) and **`is_widowed_with_children`** (1 indicates the protected group) variables. In terms of how "fairness" is encapsulated in evaluating the model, metric such as statistical parity (SP) and disparate impact (DI) are used, where SP is the difference in the probability of a positive outcome for the protected group and the non-protected group, and DI is the ratio between the proportion of positive outcome for the protected group and the non-protected group. Note that positive outcome in this context refers to people spend more money on wine products in the last two years. Given the definition, the indications of no bias is when SP = 0.0 and DI = 1.0.

Considering the priority of maximising predictive accuracy, the model 4NN has been selected subject to the fairness analysis. Resuts showcases the best model has a statistical parity of almost 0 (2.43% difference) and a disparate impact of almost 1 (0.9541). Both indicate that the bias between the protected groups, namely the less educated (below graduate level of education), and the more educated (graduate level of education and above) is not significant. Both groups have almost equal chances of being predicted as a

customer who spends more on wine products, which has defined as spending more than the median amount spent on wine products in the last two years for a given customer. Similar results (SP of 0.0353 and DI of 0.9369) has obtained when protected group is people who are widowed and with children. Therefore, based on the two metrics, it can be concluded that there is no systematic bias.

However, when compared comparing the SP and DI using the original dataset instead of the predicted values, the use of model has increased the bias where SP has increase by 0.0065 and DI decreased by 0.0105 for the less educated protected group, which requires further investigation. Given that the data has not been undergo transformation such as log transformation for better interpretability of the model, the important assumptions has been violated for models such as the LR model. Hence, it is recommended for future studies to undertake necessary transformation to satisfy the necessary assumption prior to the modelling process to ensure there is no initial bias or disadvantages to certain models, which may mitigate the limitation. Subsequently, the validity of the results in SP and DI for the refined model would be improved.

Limitations

Beyond transformation in the data processing and EDA section, outliers has not been dealt given a parsimonious approach has been adopted. Additionally, **`Education`** has been encoded randomly according to the number of unique categories exist within, which should be ordered in ways such as from basic to advanced level of education from 1 to 5, in order to produce more meaningful results. Lastly, in regards to the fairness analysis, it is recommended for future studies to undertake analysis on more protected groups in order for stakeholders to gain a more comprehensive insights into the social bias that the model have.

Conclusions

Overall, the 4NN model with Euclidean distance metric has found to be the most accurate model with the assumption that the prediction error is symmetric while acknowledging its limitation being computationally exhaustive. 4NN model revealed **`NumCatalogPurchases`**, **`NumWebPurchases`**, **`Income`**, **`NumWebVisitsMonth`**, **`AcceptedCmp5`**, **`NumStorePurchases`**, **`MntMeatProducts`**, **and `MntFishProducts`** are key driving factors behind the expenditure on wine products, with **`Income`** and **`NumCatalogPurchases`** identified as primary driving factors across different models.

References

1. McKinsey. (2020, October 5). COVID-19 digital transformation & technology | McKinsey. McKinsey & Company. https://www.mckinsey.com/capabilities/strategy-and-corporate-finance/our-insights/how-covid-19-has-pushed-companies-over-the-technology-tipping-point-and-transformed-business-forever

Appendix

Appendix 1: datatypes

```
Int64Index: 1433 entries, 502 to 1866
Data columns (total 28 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   1433 non-null   int64
 1   Year_Birth           1433 non-null   int64
 2   Education            1433 non-null   object
 3   Marital_Status       1433 non-null   object
 4   Income               1418 non-null   float64
 5   Kidhome              1433 non-null   int64
 6   Teenhome             1433 non-null   int64
 7   Dt_Customer          1433 non-null   object
 8   Recency              1433 non-null   int64
 9   MntFruits            1433 non-null   int64
 10  MntMeatProducts      1433 non-null   int64
 11  MntFishProducts      1433 non-null   int64
 12  MntSweetProducts     1433 non-null   int64
 13  MntGoldProds         1433 non-null   int64
 14  NumDealsPurchases    1433 non-null   int64
 15  NumWebPurchases      1433 non-null   int64
 16  NumCatalogPurchases  1433 non-null   int64
 17  NumStorePurchases    1433 non-null   int64
 18  NumWebVisitsMonth    1433 non-null   int64
...
 26  Z_Revenue            1433 non-null   int64
 27  Response             1433 non-null   int64
dtypes: float64(1), int64(24), object(3)
```

Appendix 2: unique values and constant columns

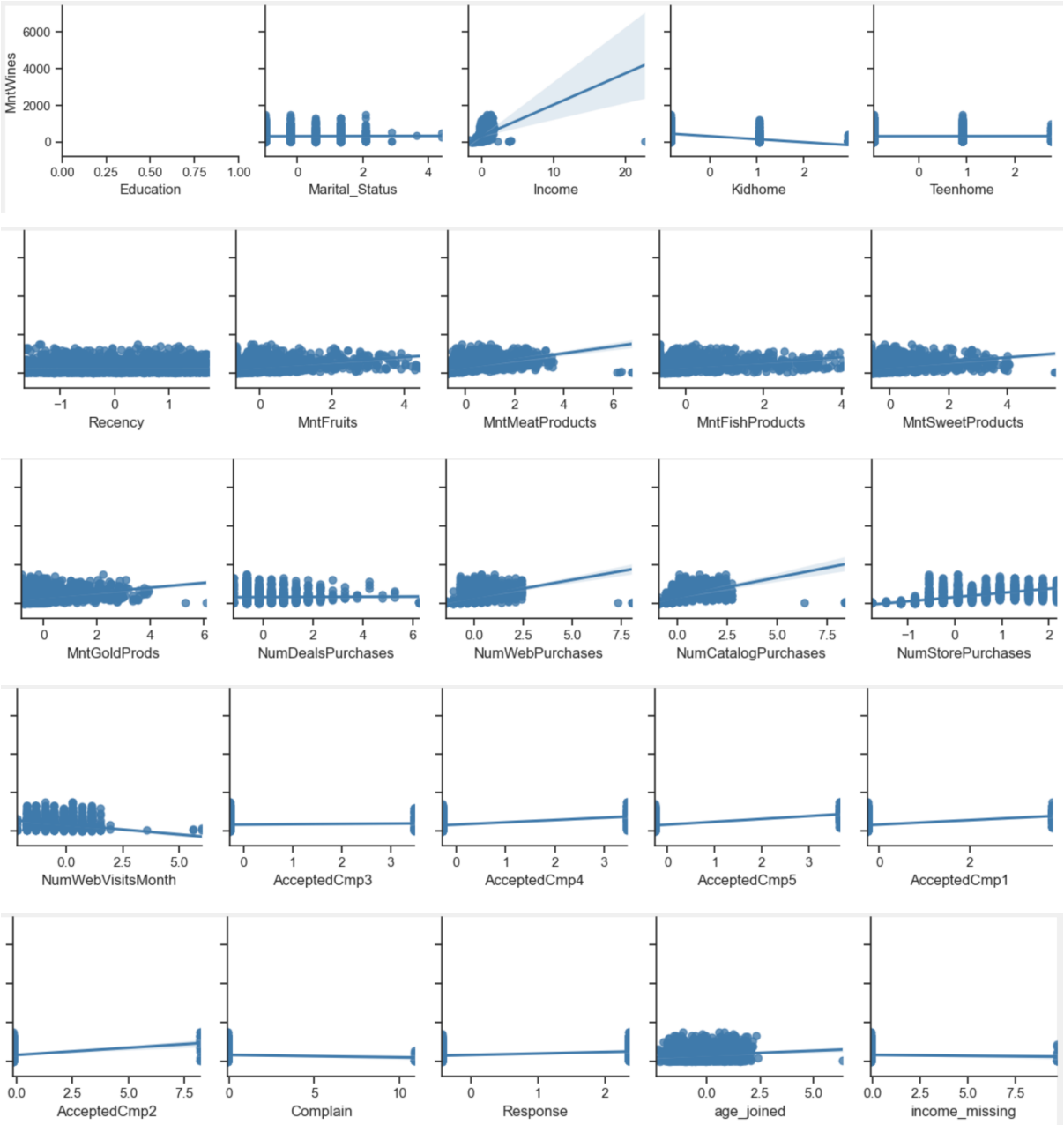
```
Constant columns:  ['Z_CostContact', 'Z_Revenue']
Invalid values:  {}
```

	Number of unique values
ID	1433
Year_Birth	57
Education	5
Marital_Status	8
Income	1312
Kidhome	3
Teenhome	3
Dt_Customer	599
Recency	100
MntFruits	146
MntMeatProducts	456
MntFishProducts	168
MntSweetProducts	159
MntGoldProds	189
NumDealsPurchases	15
NumWebPurchases	14
NumCatalogPurchases	14
NumStorePurchases	14
NumWebVisitsMonth	14
AcceptedCmp3	2
AcceptedCmp4	2
AcceptedCmp5	2
AcceptedCmp1	2
AcceptedCmp2	2
Complain	2
Z_CostContact	1
Z_Revenue	1
Response	2

Appendix 3: count of missing values

	Count of Missing Values
ID	0
Year_Birth	0
Education	0
Marital_Status	0
Income	15
Kidhome	0
Teenhome	0
Dt_Customer	0
Recency	0
MntFruits	0
MntMeatProducts	0
MntFishProducts	0
MntSweetProducts	0
MntGoldProds	0
NumDealsPurchases	0
NumWebPurchases	0
NumCatalogPurchases	0
NumStorePurchases	0
NumWebVisitsMonth	0
AcceptedCmp3	0
AcceptedCmp4	0
AcceptedCmp5	0
AcceptedCmp1	0
AcceptedCmp2	0
Complain	0
Z_CostContact	0
Z_Revenue	0
Response	0

Appendix 4: bivariate relationship between the response and the predictor



Appendix 5: MLR forward selection

MAE of the linear model that uses forward selection: 113.2355

OLS Regression Results

Dep. Variable:	MntWines	R-squared:	0.688
Model:	OLS	Adj. R-squared:	0.685
Method:	Least Squares	F-statistic:	223.5
Date:	Sat, 30 Sep 2023	Prob (F-statistic):	0.00
Time:	15:38:28	Log-Likelihood:	-9527.5
No. Observations:	1433	AIC:	1.909e+04
Df Residuals:	1418	BIC:	1.916e+04
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	305.4787	4.959	61.599	0.000	295.751	315.207
Education	-5.2754	4.984	-1.059	0.290	-15.052	4.501
Income	38.2844	6.549	5.846	0.000	25.438	51.131
MntFishProducts	-1.5747	6.741	-0.234	0.815	-14.799	11.650
MntSweetProducts	-21.0217	6.627	-3.172	0.002	-34.021	-8.022
NumDealsPurchases	-22.8625	5.833	-3.920	0.000	-34.304	-11.421
NumWebPurchases	71.8962	6.275	11.458	0.000	59.587	84.205
NumCatalogPurchases	103.2543	7.186	14.368	0.000	89.157	117.351
NumStorePurchases	117.3851	6.861	17.108	0.000	103.925	130.845
NumWebVisitsMonth	43.7080	7.351	5.946	0.000	29.287	58.129
AcceptedCmp3	7.4590	5.165	1.444	0.149	-2.672	17.590
AcceptedCmp4	46.5296	5.540	8.399	0.000	35.662	57.397
AcceptedCmp5	60.8823	5.941	10.247	0.000	49.228	72.537
AcceptedCmp1	15.6620	5.741	2.728	0.006	4.400	26.924
AcceptedCmp2	18.9215	5.305	3.567	0.000	8.516	29.327

Omnibus:	222.980	Durbin-Watson:	2.024
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1171.169
Skew:	0.615	Prob(JB):	4.83e-255
Kurtosis:	7.254	Cond. No.	3.70

Appendix 6: MLR backward selection

MAE of the linear model that uses backward selection: 113.2443

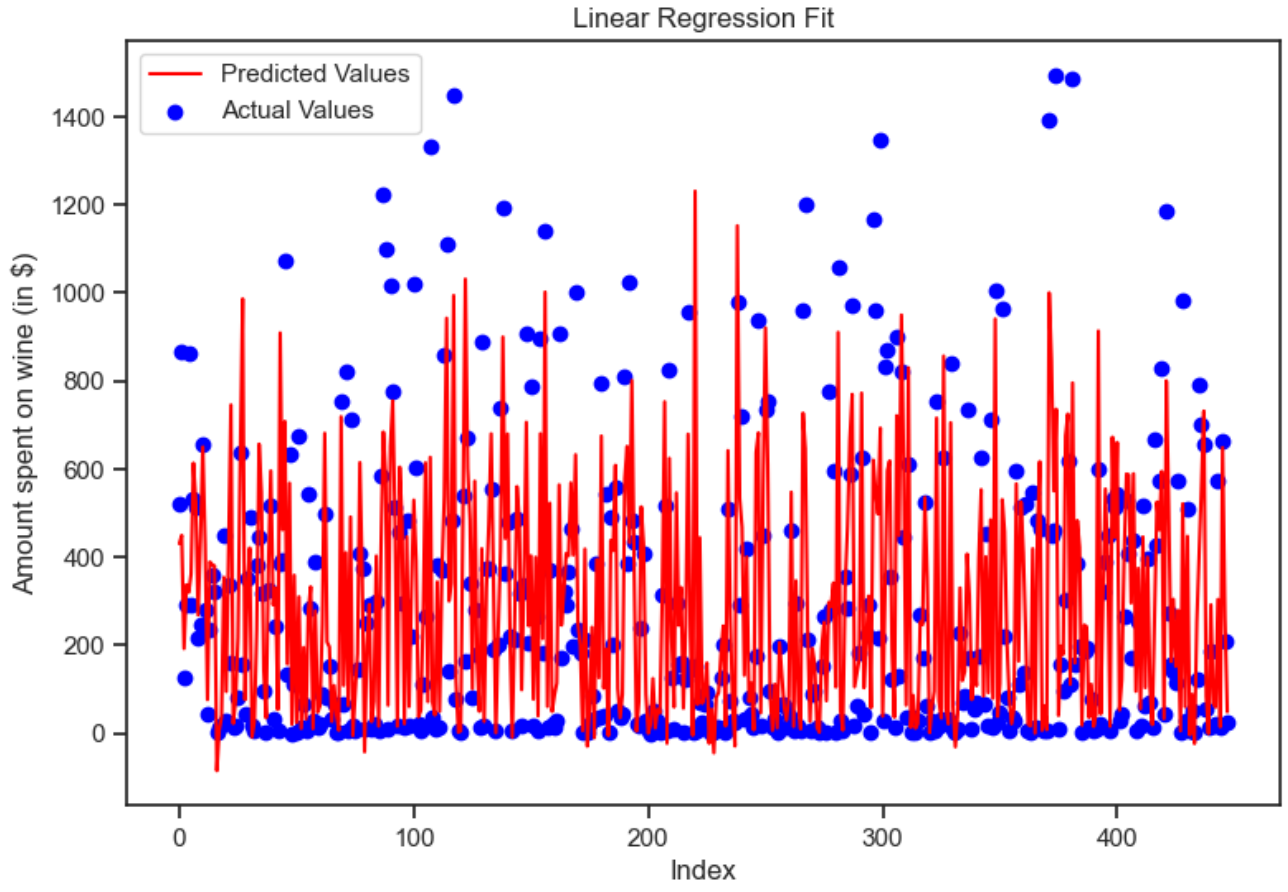
OLS Regression Results

Dep. Variable:	MntWines	R-squared:	0.690
Model:	OLS	Adj. R-squared:	0.687
Method:	Least Squares	F-statistic:	225.1
Date:	Sat, 30 Sep 2023	Prob (F-statistic):	0.00
Time:	15:39:05	Log-Likelihood:	-9524.0
No. Observations:	1433	AIC:	1.908e+04
Df Residuals:	1418	BIC:	1.916e+04
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	305.4787	4.947	61.750	0.000	295.774	315.183
Education	-5.3359	4.960	-1.076	0.282	-15.066	4.394
Income	36.5375	6.564	5.567	0.000	23.662	49.413
Teenhome	15.0103	5.664	2.650	0.008	3.899	26.122
MntSweetProducts	-19.0755	6.238	-3.058	0.002	-31.312	-6.839
NumDealsPurchases	-28.0939	6.131	-4.582	0.000	-40.121	-16.066
NumWebPurchases	69.8486	6.303	11.082	0.000	57.484	82.213
NumCatalogPurchases	105.9923	7.102	14.925	0.000	92.061	119.923
NumStorePurchases	116.1634	6.779	17.136	0.000	102.866	129.461
NumWebVisitsMonth	45.8049	7.361	6.222	0.000	31.365	60.245
AcceptedCmp3	7.1850	5.152	1.395	0.163	-2.921	17.291
AcceptedCmp4	45.7873	5.515	8.302	0.000	34.968	56.606
AcceptedCmp5	63.3256	5.959	10.628	0.000	51.637	75.014
AcceptedCmp1	16.0666	5.680	2.829	0.005	4.924	27.209
AcceptedCmp2	18.9434	5.291	3.580	0.000	8.564	29.323

Omnibus:	225.779	Durbin-Watson:	2.020
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1109.227
Skew:	0.646	Prob(JB):	1.36e-241
Kurtosis:	7.112	Cond. No.	3.59

Appendix 7: linear regression fit



Appendix 8: OLS summary of SLR

```
Best predictor:  age_joined
MAE on the validation set:  274.2544
MAE on the test set:  263.1235
Best model:                            OLS Regression Results
==============================================================================
Dep. Variable:               MntWines   R-squared:                       0.017
Model:                            OLS   Adj. R-squared:                  0.017
Method:                 Least Squares   F-statistic:                     25.45
Date:                Sat, 30 Sep 2023   Prob (F-statistic):           5.12e-07
Time:                        17:41:20   Log-Likelihood:                -10350.
No. Observations:                1433   AIC:                         2.070e+04
Df Residuals:                    1431   BIC:                         2.071e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    305.4787      8.763     34.861      0.000     288.289     322.668
age_joined    44.2077      8.763      5.045      0.000      27.018      61.397
==============================================================================
Omnibus:                      218.097   Durbin-Watson:                   2.037
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              323.731
Skew:                           1.139   Prob(JB):                     5.04e-71
Kurtosis:                       3.482   Cond. No.                         1.00
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```