MPhil in Economics and Data Science

Module: D100/D400

Candidate Number (BNG): 3384D

Deadline date: 19 Dec 2024

I confirm that this is entirely my own work and has not previously been submitted for assessment, and I have read and understood the University's and Faculty's definition of Plagiarism

Actual word count: 1977

Introduction

Accurately predicting customer income is crucial for businesses aiming to tailor their marketing strategies and product offerings. This report focuses on developing a predictive model for customer income using a detailed dataset containing demographic, behavioral, and transactional variables.

The dataset includes 28 variables:

- **Demographics**: Year of birth, education level, marital status, and yearly household income.
- **Household Composition**: Number of children (Kidhome) and teenagers (Teenhome) in the household.
- **Engagement**: Number of days since the last purchase (Recency) and whether the customer lodged a complaint in the past two years (Complain).
- **Spending Habits**: Amount spent in the past two years on wine, fruits, meat, fish, sweets, and gold products.
- **Promotion Response**: Number of purchases made with discounts (NumDealsPurchases) and responses to various marketing campaigns.
- **Purchasing Channels**: Number of purchases made through the website, catalogs, and in stores, as well as the number of website visits in the last month.

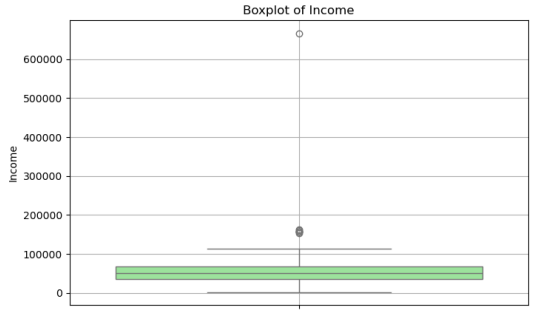
By leveraging these variables, this project aims to build GLM and GBM models to accurately predict income. In the next part, Explanatory Data Analysis (EDA) and then feature selection, GLM and GBM model building and their evaluations will be presented.

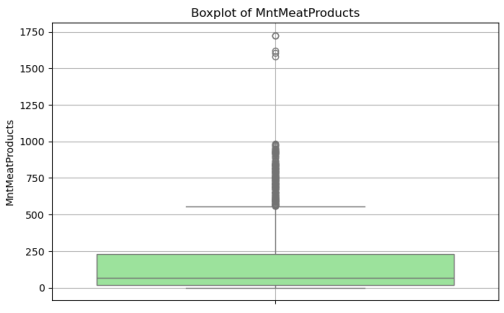
Explanatory Data Analysis (EDA)

Firstly we checked the data summary and the data type. We found that most of the variables are numerical, only 'Education', 'Marital_Status' and ''Dt_Customer' are type object.

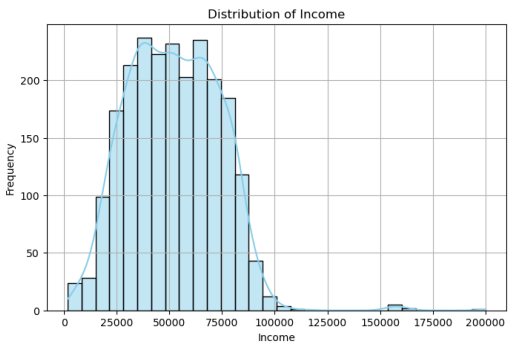
Then we checked the missing values in the dataset and found out we had 24 missing values for 'Income', our target variable.

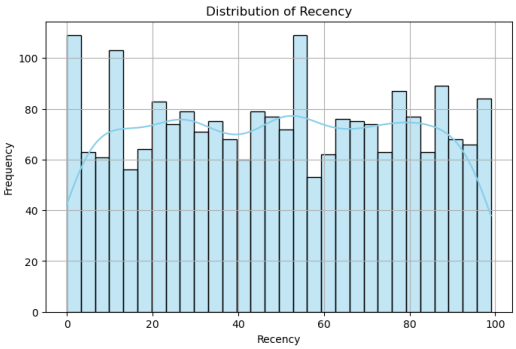
We focus on numerical variables first, plotting the boxplots for 'Income' and 'MntMeatProducts'. Some extreme values have been found.



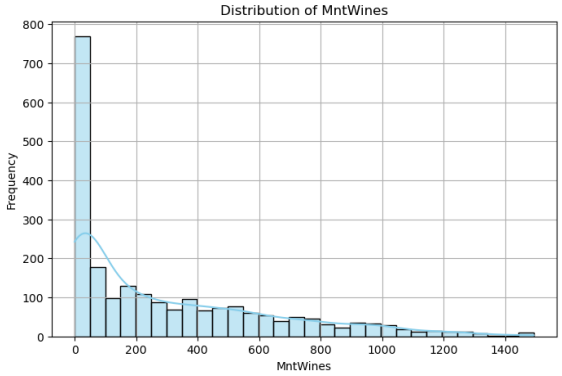


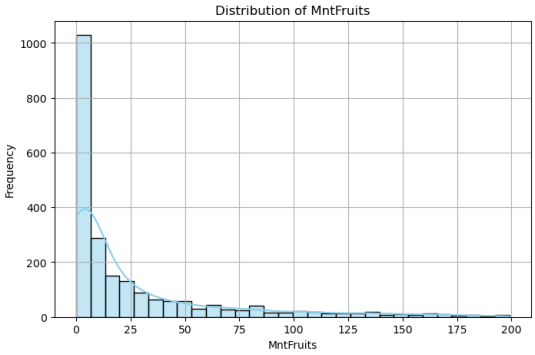
Next we plotted histograms for 'Income' and 'Recency', to discover their distributions. We found that 'Income' has a likely Gaussian distribution, and 'Recency' has a more uniform distribution. Distribution for 'Income' could be a standard in choosing our GLM model, and Skewness of 'Income' has been calculated to be 0.473.



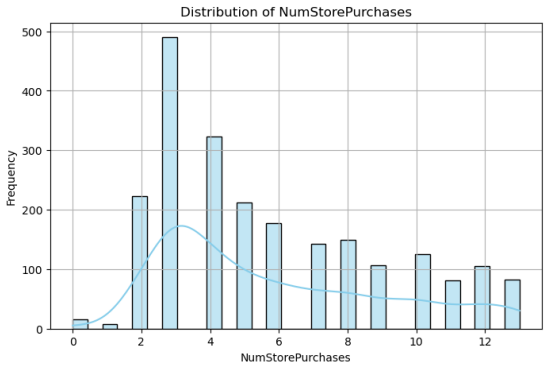


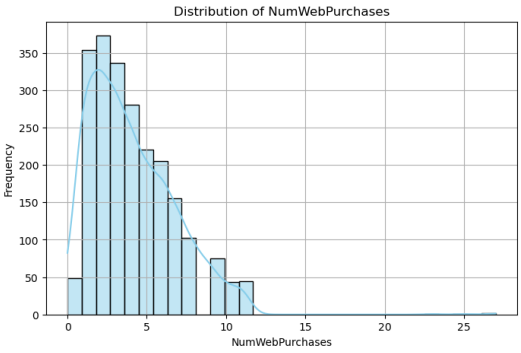
Next distributions of 'MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts' and 'MntGoldProds' are plotted, and they are all largely right skewed. Here we only show two of the histograms.



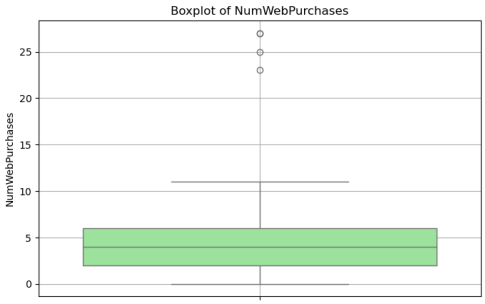


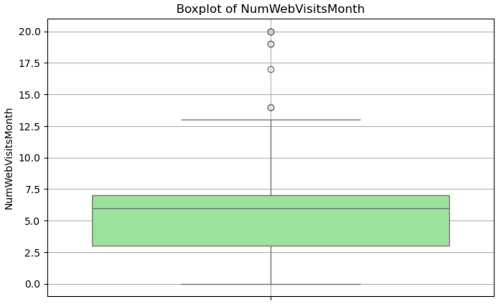
Other than large numerical continuous variables, there are also some numerical variables with smaller integer numbers, such as 'NumStorePurchases' and 'NumWebVisitsMonth', 'NumDealsPurchases' and 'NumWebPurchases'. Here are two of the distributions:



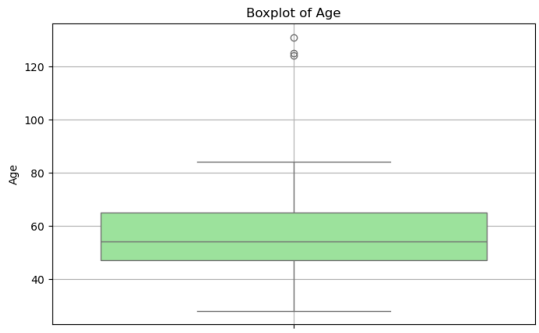


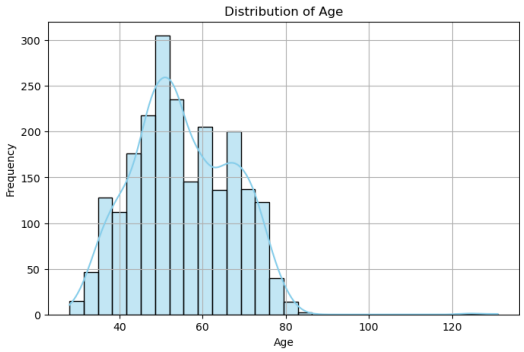
Some of these variables have extreme values, which could be outliers, so we analysed their boxplots:



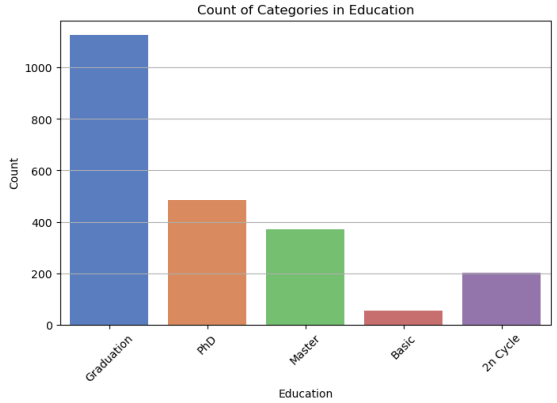


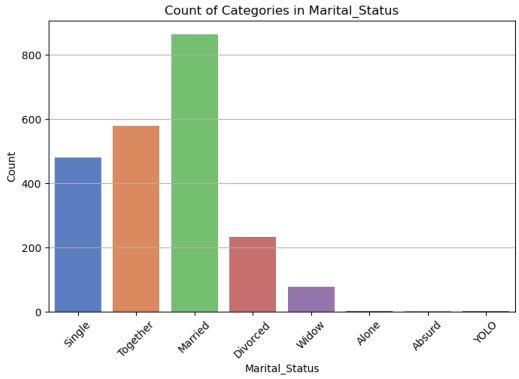
Afterwards, we transformed ‘Year_birth’ to a new variable ‘Age’, which could be useful for further transformations in the model, this variable seems to have a normal distribution, with some extreme values:





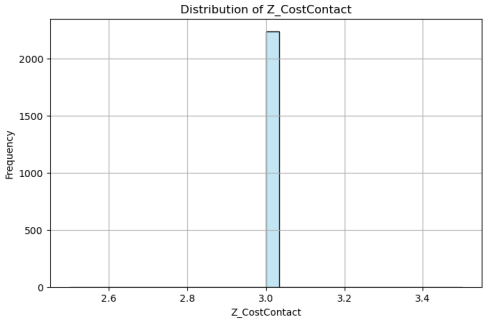
Now we focus on categorical variables, ‘Education’ and ‘Marital_Status’:



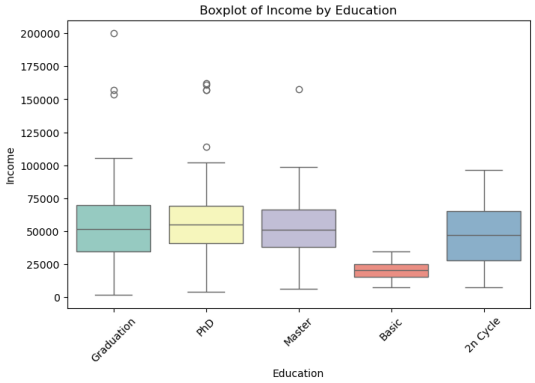


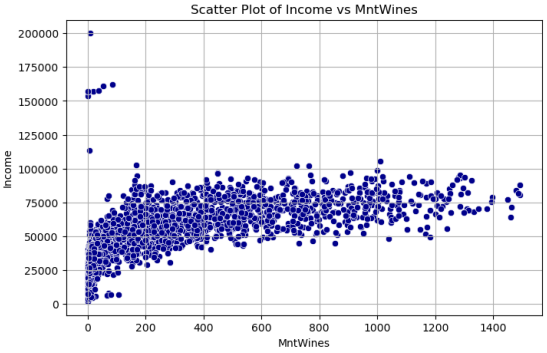
It can be seen that there are few values in ‘Marital_Status’ that are extreme.

There are also some binary variables, such as ‘AcceptedCmp1’, ‘AcceptedCmp2’, etc. The definitions of them are not suitable for income so we will not be using them in the model. There are also some constant variables, with no definition, such as ‘Z_CostContact’, they will also not be used:



After analysing distributions of the variables, we can plot some scatter plots to see whether some features are correlated to the target variable ‘Income’, here we show one categorical and one numerical variable with the targeting variable:



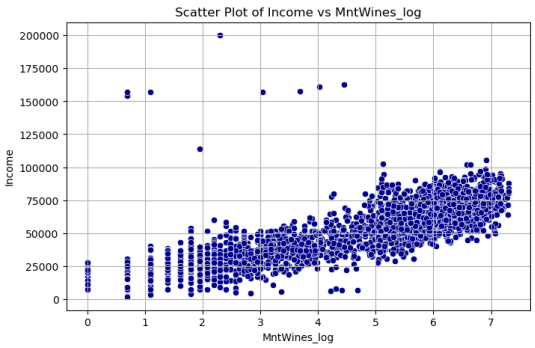


We can see that 'Education' has some correlation with the target variable, and 'MntWines' also shows some positive relationship with it, but not very clear.

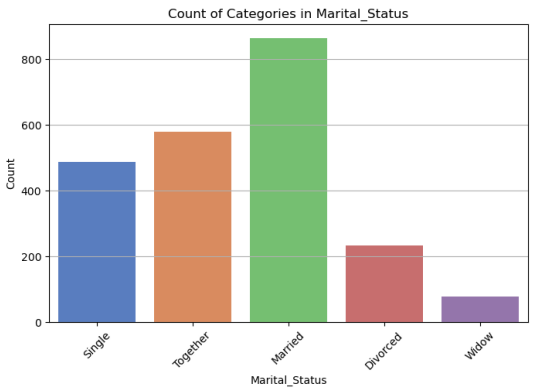
Next step we will do data cleaning, but using combining, clipping, log_transformation and binning:

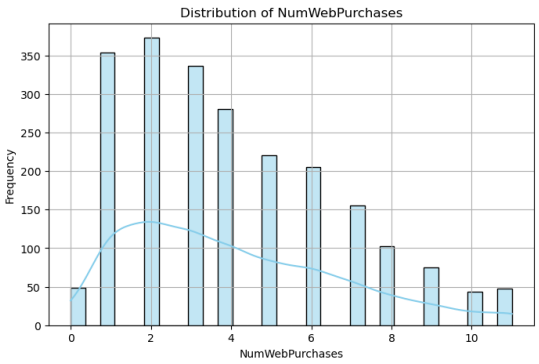
- log_transformation for right skewed variables, such as 'MntWines', 'MntFruits', etc
- combining 'KidHome' and 'TeenHome', create a new feature: 'Children'
- in 'Marital_Status', "alone", "absurd" and "yolo" seem to have relatively small amount, and the definition of these are not clear, so I combine them into 'Single'
- make upper clip for variables with extreme values, such as 'Income', 'NumWebPurchase' and 'NumWebVisitsMonth'
- for the new variable 'Age', I did binning, where bins=[31, 41, 51, 61,71] and now 'Age' becomes a categorical variable

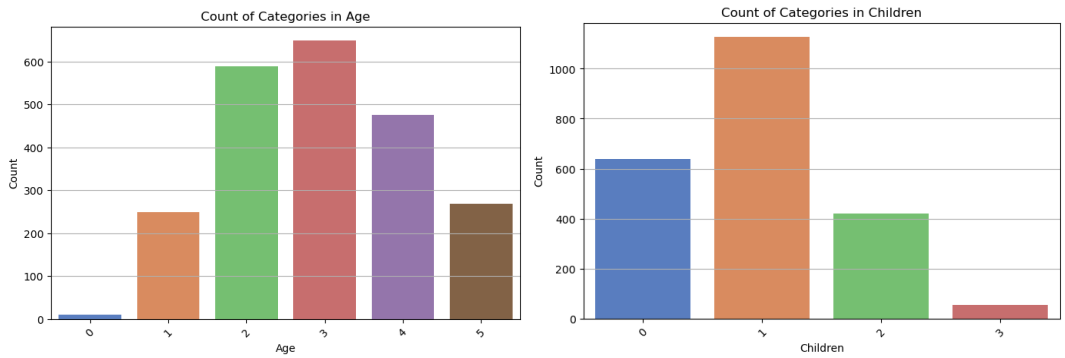
Log-transformed variables showed more positive relationships with targe variable:



Here are other distributions from other cleaned features:







After cleaning data, the features that we are using are:
"Education", "Marital_Status", "Age","Children", "NumStorePurchases", "NumWebPurchases", "NumWebVisitsMonth","NumDealsPurchases", "MntWines_log", "MntFruits_log", "MntMeatProducts_log", "MntGoldProds_log", "MntFishProducts_log".

Feature Selection and Engineering

Feature Selection

Feature selection was driven by domain knowledge and exploratory data analysis (EDA). The primary goal was to identify predictors most relevant to the income prediction task while avoiding multicollinearity and irrelevant variables. Key decisions included:

Categorical Features:

- Variables like 'Education', 'Marital_Status', and 'NumWebVisitsMonth' were included due to their potential influence on spending behaviors and, indirectly, income levels. We included 'Children' and binned 'Age' to be categorical because they have relatively small numbers and can
- Other categorical variables, such as 'NumStorePurchases' and 'NumDealsPurchases', were selected for their direct correlation with income observed during EDA.

Numerical Features:

- Spending-related variables (MntWines_log, MntFruits_log, etc.) were retained as they showed significant predictive power in the initial analysis.

Feature Engineering

To enhance the predictive power of the models, several feature engineering techniques were applied in the pipelines of my models:

Spline Transformations:

- To account for non-linear relationships between numerical features and income, splines were introduced via the SplineTransformer. This transformation enabled the models to capture subtle variations without assuming strict linearity. Numerical features such as "MntFishProducts_log" may have non-linear relationship with 'Income'. We chose knots="quantile", which places the knots at the quantiles of the feature distribution, ensuring even coverage.

Encoding Categorical Variables:

- We used OneHotEncoder to create binary columns for each category while dropping the first category to avoid multicollinearity in the models. Encoded features could also avoid imposing ordinal assumptions. This approach ensured that models could weigh each category independently.

Model improvement and evaluation

Evaluation Approach

Firstly, we split the dataset into training and testing sets to ensure that the model's performance could be validated on unseen data. The primary evaluation metric was Mean Absolute Error (MAE), chosen for its interpretability and robustness against outliers.
For GLM, we chose gaussian and gamma families for the models and compared their MAE, because the target variable is slightly right skewed but mostly look like gaussian distribution. Gaussian GLM has a lower MAE so we sticked to this family.
Then we tried hyperparameter tuning for GLM model, using pipelines where feature engineering is used and cross-validation, to find lowest MAE.
For GBM, we used different pipelines and cross-validation, but still tried to find a lower MAE.
In the end, we used Lorenz curves, SHAP values and partial dependence plots to find feature importance and model behavior.

Hyperparameter Tuning

GLM Tuning

Hyperparameter tuning for GLMs focused on regularization parameters:

- **Alpha**: This parameter controls the overall strength of regularization. A higher value increases regularization, simplifying the model but potentially underfitting.
- **L1 Ratio**: This parameter sets the proportion between L1 (Lasso) and L2 (Ridge) penalties, controlling how sparsity and coefficient shrinkage are enforced.

After choosing the parameters, we used GridSearchCV to performe 10-fold cross-validation to ensure robust evaluation. We chose tested values: alpha: [0.01, 0.1, 1]
l1_ratio: [0, 0.25, 0.5, 0.75, 1]. We also created a scoring function in GridSearchCV to ensure a small MAE is found.

GBM Tuning

Gradient Boosting Machines were optimized for the following parameters:

- **Learning Rate**: Controlled the step size during gradient descent.
- **Number of Estimators**: Defined the maximum number of boosting iterations, with early stopping used to terminate training when no significant improvements were observed.
- **Number of Leaves**: Dictated the complexity of each tree, balancing model flexibility and overfitting risk.
- **Minimum Child Weight**: Ensured that leaf nodes had sufficient weight to avoid overfitting smaller splits.

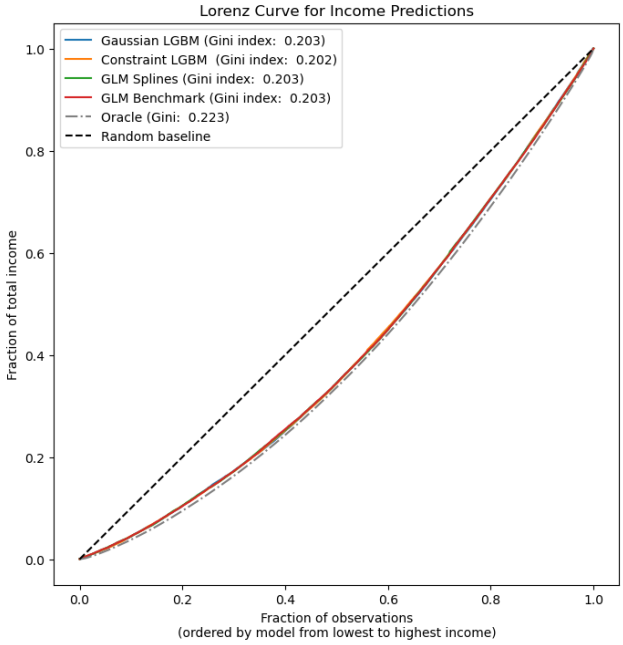
- **Monotonic Constraints**: Applied to enforce logical relationships between features and income.

We firstly tuned unconstrained LGBM model, by the first 4 parameters, and then tuned monotonic-constrained LGBM model. We chose monotonic constraints by the results in EDA

where we saw the relationship between each numerical features with the target variable.
In this part, we also used a GridSearchCV approach with 5-fold cross-validation, testing the following ranges: Learning Rate: [0.01, 0.02, 0.05, 0.1], Number of Leaves: [6, 12, 24], Minimum Child Weight: [1, 5, 10] and set Number of Estimators: 1000, the same scorer was used in the tuning process.

Comparing results

By tuning GLM models, including Alpha L1 ratio, MAE has decreased from 6629 to 6595. And the best parameters from tuning are {'estimate__alpha': 1, 'estimate__l1_ratio': 1}
By tuning GBM, including learning rate, Number of leaves, Minimum Child Weight and finally Monotonic Constraints, MAE from tuned unconstrained LGBM has decreased from 6527.062 to 6485.244. The parameters with the best results are {'estimate__learning_rate': 0.05, 'estimate__min_child_weight': 1, 'estimate__n_estimators': 1000, 'estimate__num_leaves': 6}. MAE of tuned constrained LGBM has a lower MAE of 6343.21, with parameters {'estimate__learning_rate': 0.01, 'estimate__min_child_weight': 1, 'estimate__n_estimators': 1000, 'estimate__num_leaves': 12}

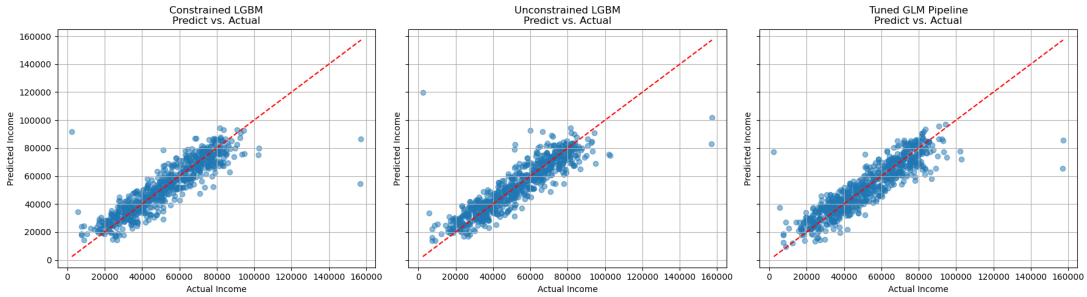


From the Lorenz Curve for each model, we can see that:

- The Oracle model represents perfect predictions, where all income distribution is correctly captured. This serves as the upper bound for predictive accuracy and provides a benchmark for the other models.
- Constrained and Unconstrained GBM: Both models show similar Gini indices (approximately 0.202-0.203), indicating comparable performance in predicting income distribution. The constrained GBM slightly underperforms compared to the unconstrained GBM (0.202 vs. 0.203), likely due to the added monotonic constraints, which prioritize interpretability and logical consistency over absolute accuracy. The flexibility in monotonic constraints have been decreased.
- GLM Models: GLM Splines and GLM Benchmark perform similarly, with a Gini index of 0.203. This indicates that spline transformations and OneHotEncoding did not make a

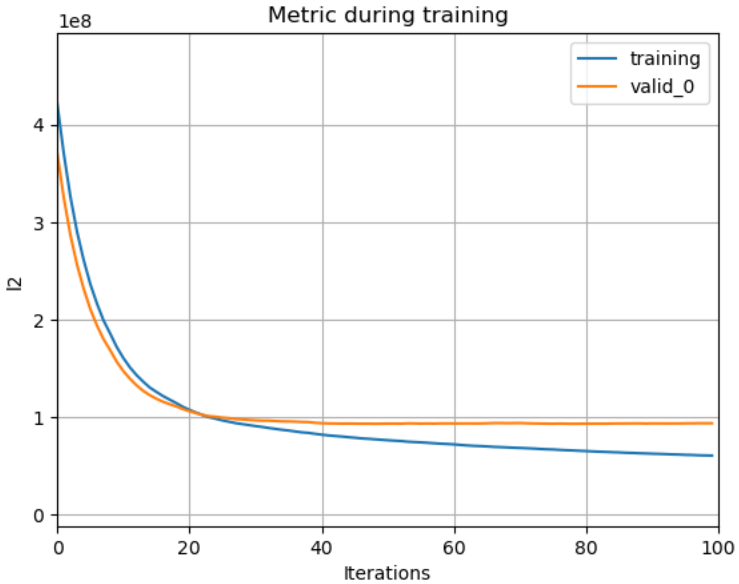
big impact on improving model's ability. It might because of the categorical features were not suitable for the encoding.

By plotting 'Predicted vs Actual' plot, we can see that all three models have a similar quality and there are deviations around the diagonal, indicating they struggled to make perfect predictions among all income range. The decrease from MAE is not well virsualised here, meaning improvements among models are limited. This might because of the misuse of features, as they all share the same ones. The used features may not fully explain the target variable. This result aligns with Lorenz curve.

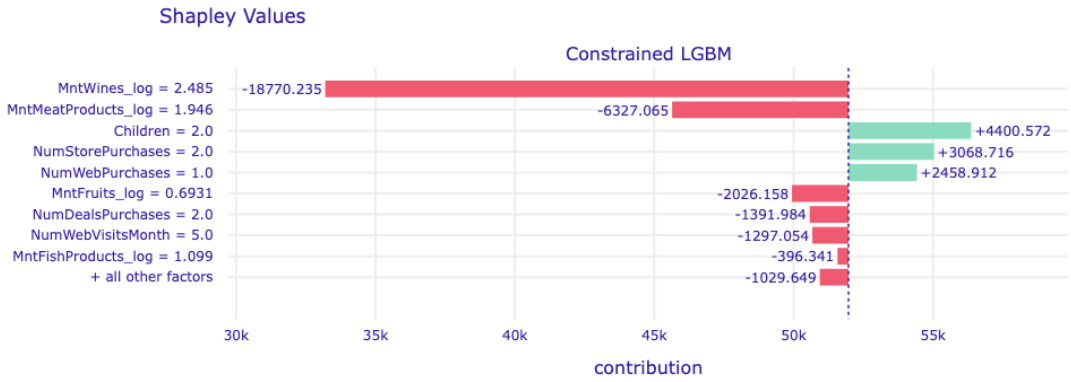


Performance of Final model

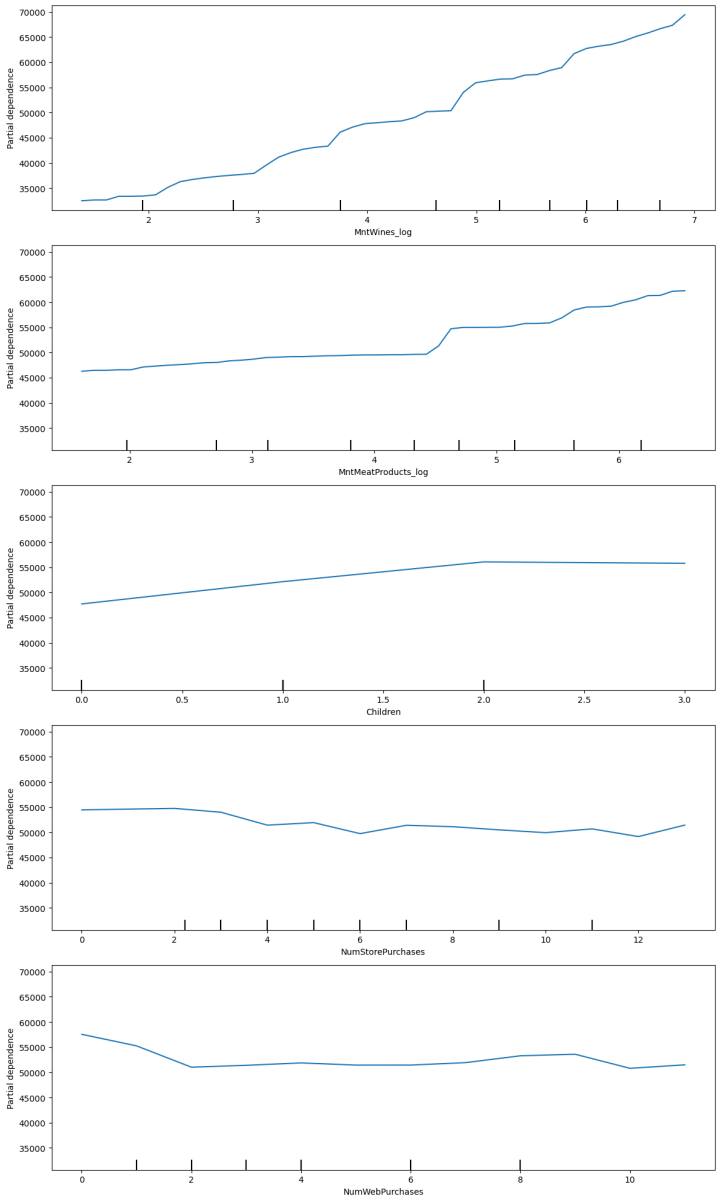
Here is the learning curve of the constrained LGBM. We can see a steep initial decline, showing a quick learning. After about 40 iterations validation curve becomes flat while training curve still decreases, but their gap is still not large, indicating no significant overfitting, possibly because of the monotonic constraints. We can use early stop at around 40 to prevent unnecessary computation.



Here is the Shapley Values plot of the constrained LGBM. We have top five features where 'MntWines_log' and 'MntMeatProducrs_log' have the largest negative contributions to the target variable (-18770 and -6327), while 'Children', 'NumStorePurchases' and 'NumWebPruchases' have some positive impacts on 'Income'.



We then plotted partial dependency plots, in contradiction, the first two features showed a positive dependency with 'Income', suggesting they generally increases 'Income', but might be affected by other features. 'Children' still has positive effect on 'Income', while the last two features seem flat and weak, which might be due to other features' impacts.



Conclusion and Outlook

By data-cleaning, feature selection and engineering, useful features are extracted to be used in the following models. By tuning GLM Gaussian model, unconstrained and constrained LGBM models, we obtained better MAE across the process and we obtained the best model: constrained LGBM. However, the prediction performance is not ideal as expected, with similar and small gini-index, deviation in actual and predicted values among the models, indicating some improvements.

The model would be improved if some geographical and more behavioral data are provided, such as region and employment status. Other than data, advanced feature engineering could be applied, such as making interactions (eg, 'NumStorePurchases'*'Age'). Some unnecessary features should also be dropped (based on SHAP values). The evaluation could be improved by involving more standards other than MAE. More complex model could be used such as neural networks (if have more time).