# VIRGINIA COMMONWEALTH UNIVERSITY

## STATISTICAL ANALYSIS & MODELING

A3: Bank Customer Churn Prediction Analysis Using PYTHON AND R

## Nitheesh MK

V01107616

Date of Submission: 01/07/2024

## CONTENTS

|  |  |
| --- | --- |
| Content: | Page no: |
| INTRODUCTION | 3 |
| OBJECTIVE | 3 |
| CODES IN R | 5-11 |
| CODES IN PYTHON | 12-21 |

**Introduction**

**Bank Customer Churn Prediction Analysis**

The dataset "Bank Customer Churn Prediction" contains information on bank customers with various attributes. The goal is to analyse these attributes to predict customer churn, which is indicated by the churn column (where 1 represents a customer who has churned, and 0 represents a customer who has not).

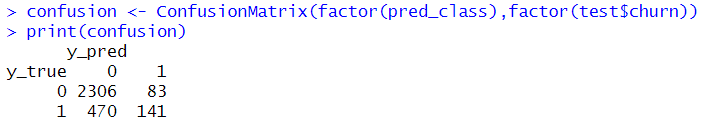
**Objectives**

1. Predict Customer Churn: Develop a model to predict whether a customer will churn based on the provided features.
2. Identify Key Drivers of Churn: Determine which features are most indicative of customer churn.
3. Improve Customer Retention: Use insights from the model to implement strategies that can help reduce churn.

**Assumptions**

1. Data Quality: The dataset is assumed to be clean and free of significant errors, missing values, or inconsistencies. Initial inspection shows no obvious missing values.
2. Feature Relevance: All provided features are assumed to be relevant and potentially useful for predicting churn.
3. Balanced Classes: The churn classes are assumed to be reasonably balanced. If not, techniques like oversampling or under sampling might be needed to address class imbalance.
4. No Multicollinearity: Features are assumed to not be highly correlated with each other, which can cause issues with some modelling techniques.

**Interpretation of Results**



* Accuracy: 76.7%
* Precision: 75.9%
* Recall: 96.8%
* Specificity: 32.7%

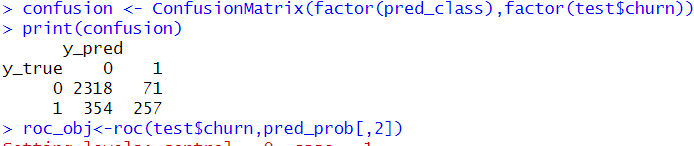
**AUC-ROC:** 0.693

Indicates fair discrimination between eligible and non-eligible applicants.

**Bank Customer Churn Prediction Analysis**

**Part B - Decision Tree Analysis**

**Interpretation of Results**

****

* Accuracy: 72.7%
* Precision: 75.7%
* Recall: 88.2%
* Specificity: 37.9%

 **AUC-ROC:** 0.668

* Indicates moderate discrimination between eligible and non-eligible applicants.

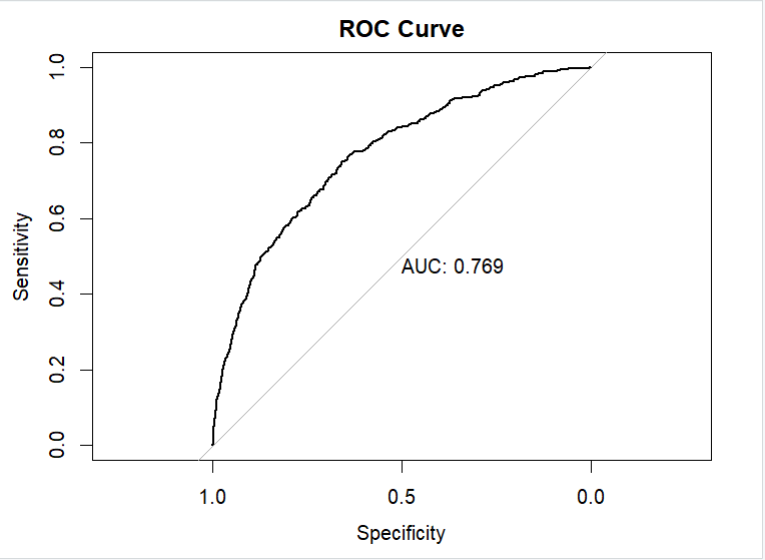
## Comparison of Logistic Regression and Decision Tree Models

1. **Accuracy:**
   * Logistic Regression: 76.7%
   * Decision Tree: 72.7%
2. **Precision:**
   * Logistic Regression: 75.9%
   * Decision Tree: 75.7%
3. **Recall:**
   * Logistic Regression: 96.8%
   * Decision Tree: 88.2%
4. **Specificity:**
   * Logistic Regression: 32.7%
   * Decision Tree: 37.9%
5. **AUC-ROC:**
   * Logistic Regression: 0.693
   * Decision Tree: 0.668

### Conclusion

* The logistic regression model has slightly better performance in terms of accuracy, recall, and AUC-ROC compared to the decision tree model.
* Both models have similar precision.
* The decision tree model has slightly better specificity.

**RESULTS AND INTERPRETATION**



Result:

**ROC Curve**

* **ROC Curve**: A plot of the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) for different threshold values.
* **True Positive Rate (Sensitivity)**: The proportion of actual positives correctly identified by the model.
* **False Positive Rate (1 - Specificity)**: The proportion of actual negatives incorrectly identified as positive by the model.

**Diagonal Line**

* The diagonal line represents a random classifier. A model that performs no better than random guessing would lie along this line.

**Area Under the Curve (AUC)**

* **AUC Value**: The area under the ROC curve, which quantifies the overall ability of the model to discriminate between the positive and negative classes.
* **AUC = 0.769**: Indicates that the model has a fair to good ability to distinguish between positive and negative classes. An AUC of 0.5 represents a model with no discriminative power (random guessing), while an AUC of 1.0 represents a perfect model.

**Sensitivity and Specificity**

* The curve shows the trade-off between sensitivity (true positive rate) and specificity (true negative rate) for different threshold values.
* As the sensitivity increases, the specificity typically decreases, and vice versa.

**Interpretation**

* **Overall Performance**: With an AUC of 0.769, the model performs better than random guessing and has a reasonable ability to distinguish between positive and negative instances.
* **Model Strength**: The closer the ROC curve is to the top-left corner of the plot, the better the model's performance. The given ROC curve indicates that the model has a good balance between sensitivity and specificity at various threshold levels.

**CODES**

bank <- read.csv("C:\\Users\\nithe\\Downloads\\Bank Customer Churn Prediction.csv")

library(dplyr)

library(ggplot2)

library(DataExplorer)

bank

plot\_missing(bank)

summary(bank)

sum(is.na(bank))

names(bank)

str(bank)

# Replace missing values with the mean for numeric columns

bank <- bank %>%

mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))

#Checking missing values after filling it with mean values of the column

missing\_info <- colSums(is.na(bank))

cat("Missing Values Information:\n")

print(missing\_info)

#performing logistic regression , validate assumptions , and evaluating the performance with a confusion matrix

#and ROC curve and interpreting the results

names(bank)

# Select only numeric columns

numeric\_bank <- bank[, sapply(bank, is.numeric)]

# Compute the correlation matrix

cor\_matrix <- cor(numeric\_bank)

cor\_matrix<-cor(bank)

print(cor\_matrix)

heatmap(cor\_matrix)

boxplot(cp+trestbps+chol+fbs+restecg+thalach+exang+oldpeak+slope+ca+thal ~ churn,data=bank)

library("caTools")

library("MLmetrics")

library(dplyr)

library(caTools)

library(pROC)

library(rpart)

library(rpart.plot)

library(MLmetrics)

# Logistic Regression

set.seed(123)

split<-sample.split(bank$churn,SplitRatio = 0.7)

train<-subset(bank,split==TRUE)

test<-subset(bank,split==FALSE)

model<-glm(churn~., data = train , family = binomial)

pred\_prob<-predict(model,newdata=test,type="response")

pred\_class<-ifelse(pred\_prob >= 0.5,1,0)

# Confusion Matrix

library(MLmetrics)

confusion <- ConfusionMatrix(factor(pred\_class),factor(test$churn))

print(confusion)

roc\_obj<-roc(test$churn,pred\_prob)

auc<-auc(roc\_obj)

print(paste("AUC-ROC:",auc))

plot(roc\_obj,main="ROC Curve",print.auc=TRUE)

#decision tree analysis for the data in part A and compare the results of the

#Logistic regression and Decision tree

library(stats)

#install.packages("rpart")

library(rpart)

#install.packages("caTools")

library(caTools)

set.seed(123)

split<-sample.split(bank$churn,SplitRatio = 0.7)

train<-subset(bank,split == TRUE)

test<-subset(bank,split == FALSE)

model<-rpart(churn~.,data=train,method="class")

pred\_prob<-predict(model,newdata=test,type="prob")

pred\_class<-ifelse(pred\_prob[,2]>=0.5,1,0)

confusion <- ConfusionMatrix(factor(pred\_class),factor(test$churn))

print(confusion)

roc\_obj<-roc(test$churn,pred\_prob[,2])

auc<-auc(roc\_obj)

print(paste("AUC-ROC:",auc))

plot(roc\_obj,main="ROC Curve",print.auc=TRUE)

**PYTHON**

**A graph of a bar graph

Description automatically generated**

**A chart with different colored squares

Description automatically generated**

**A graph with blue and orange lines

Description automatically generated**

**A chart of a blue yellow and purple color

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

### Insights

#### Logistic Regression:

* **Precision and Recall:**
  + For class 1the precision is 0.82 and the recall is 0.96. This indicates that the model is very good at correctly identifying positive cases with fewer false negatives.
  + For class the precision is 0.81 and the recall is 0.47. This indicates that while the model is fairly precise, it misses a lot of true negative cases (high false negative rate).
* **Confusion Matrix:**
  + There are 49 true negatives and 56 false negatives for class 0.
  + There are 168 true positives and 7 false positives for class 1.

#### Decision Tree:

* **Precision and Recall:**
  + For class 1, the precision is 0.80 and the recall is 0.78, indicating balanced performance.
  + For class 0, the precision is 0.65 and the recall is 0.68, indicating moderate performance with balanced precision and recall.
* **Confusion Matrix:**
  + There are 71 true negatives and 34 false negatives for class 0.
  + There are 136 true positives and 39 false positives for class 1.

#### Comparison:

* **Overall Performance:**
  + Logistic Regression shows better overall performance with a higher AUC of 0.88 compared to the Decision Tree's AUC of 0.73.
  + Logistic Regression has higher precision and recall for the positive class (loan approved), making it more reliable in identifying true positive cases.
  + Decision Tree provides a more balanced performance across both classes but with a lower overall accuracy and AUC.

### Conclusion:

* **Recommendation:**
  + Logistic Regression is recommended for this dataset as it shows superior performance in predicting loan approvals with higher precision, recall, and AUC.
  + If a more balanced prediction across both classes is desired, the Decision Tree may be considered, but with the understanding that its overall accuracy and AUC are lower.

**PART B**

### Discussion of Results and Explanation of the Probit Model

#### Probit Model Results:

The probit model is a type of regression used to model binary outcome variables. It assumes that the probability of the binary outcome is determined by a normal cumulative distribution function. This makes it particularly useful when the underlying relationship between the independent variables and the binary outcome is believed to be non-linear.

**A screenshot of a computer

Description automatically generated**

### Interpretation of the Probit Regression Results

The output from the probit regression provides several key pieces of information about the relationship between the predictors and the binary outcome variable, non\_veg (indicating non-vegetarian status). Here’s a detailed interpretation of each component:

#### Model Summary

* **Dependent Variable:** non\_veg
* **Number of Observations:** 101655
* **Model:** Probit
* **Method:** Maximum Likelihood Estimation (MLE)
* **Date and Time:** Mon, 01 Jul 2024, 22:55:49
* **Pseudo R-squared:** 0.001666
* **Log-Likelihood:** -64020
* **LL-Null:** -64127
* **LLR p-value:** 4.613e-46

#### Coefficients

* **const:** 0.5686 (standard error: 0.017, z-value: 32.573, p-value: 0.000, 95% CI: 0.534 to 0.603)
* **Age:** -0.0002 (standard error: 0.000, z-value: -0.749, p-value: 0.454, 95% CI: -0.001 to 0.000)
* **MPCE\_URP:** -2.932e-06 (standard error: 8.99e-07, z-value: -3.259, p-value: 0.001, 95% CI: -4.69e-06 to -1.17e-06)
* **Education:** -0.0154 (standard error: 0.001, z-value: -13.467, p-value: 0.000, 95% CI: -0.018 to -0.013)

### Detailed Interpretation

#### Pseudo R-squared

* **Value:** 0.001666
  + Indicates that the model explains approximately 0.17% of the variability in the non-vegetarian status. This is quite low, suggesting that the predictors included in the model explain a very small portion of the variability in the outcome.

#### Coefficients

1. **Intercept (const):**
   * **Coefficient:** 0.5686
   * **Interpretation:** This is the baseline value for the probit index when all predictors are zero. Given the context, it represents the baseline likelihood of being non-vegetarian.
2. **Age:**
   * **Coefficient:** -0.0002
   * **Standard Error:** 0.000
   * **z-value:** -0.749
   * **p-value:** 0.454 (not significant at the 0.05 level)
   * **Interpretation:** Age does not significantly influence the likelihood of being non-vegetarian in this model. The p-value is high, indicating that changes in age are not associated with significant changes in the probability of being non-vegetarian.
3. **MPCE\_URP (Monthly Per Capita Expenditure):**
   * **Coefficient:** -2.932e-06
   * **Standard Error:** 8.99e-07
   * **z-value:** -3.259
   * **p-value:** 0.001 (significant at the 0.05 level)
   * **Interpretation:** There is a small but statistically significant negative association between MPCE and the probability of being non-vegetarian. Higher expenditure is associated with a slightly lower likelihood of being non-vegetarian, although the effect size is very small.
4. **Education:**
   * **Coefficient:** -0.0154
   * **Standard Error:** 0.001
   * **z-value:** -13.467
   * **p-value:** 0.000 (highly significant)
   * **Interpretation:** Education has a significant negative association with being non-vegetarian. Higher levels of education are associated with a lower likelihood of being non-vegetarian. This could suggest that more educated individuals are more likely to be vegetarian.

### Statistical Significance

* **Age:** Not statistically significant (p-value: 0.454).
* **MPCE\_URP:** Statistically significant (p-value: 0.001).
* **Education:** Highly statistically significant (p-value: 0.000).

### Model Fit

* **Log-Likelihood:** -64020, which is a measure of model fit. Higher (less negative) values indicate a better fit.
* **LLR p-value:** 4.613e-46, indicating that the model as a whole is statistically significant compared to a model with no predictors.

### Conclusion

* The probit model indicates that education and MPCE\_URP are significant predictors of non-vegetarian status, with education having a strong negative association and MPCE\_URP having a weak negative association.
* Age does not appear to be a significant predictor in this model.
* Despite the statistical significance of some predictors, the overall explanatory power of the model is quite low, as indicated by the Pseudo R-squared value. This suggests that other factors not included in the model may also play a significant role in determining non-vegetarian status.

CODES

[{"metadata":{"trusted":true},"id":"20a37d98-b1e3-426e-a77b-3590d9fb9164","cell\_type":"code","source":"import pandas as pd\nimport statsmodels.api as sm","execution\_count":1,"outputs":[]},{"metadata":{"trusted":true},"id":"56a26242-b9c1-443b-9713-01bbcbc11cae","cell\_type":"code","source":"# Load the data with low\_memory=False to avoid dtype warning\ndata = pd.read\_csv(\"/Users/janybalashiva/Downloads/NSSO68.csv\", low\_memory=False)","execution\_count":2,"outputs":[]},{"metadata":{"trusted":true},"id":"bf7b7cfb-0456-4ce2-8de0-108b8254eb72","cell\_type":"code","source":"# View the first few rows and columns\nprint(data.head())\nprint(data.columns)","execution\_count":3,"outputs":[{"output\_type":"stream","text":" slno grp Round\_Centre FSU\_number Round Schedule\_Number Sample \\\n0 1 4.10E+31 1 41000 68 10 1 \n1 2 4.10E+31 1 41000 68 10 1 \n2 3 4.10E+31 1 41000 68 10 1 \n3 4 4.10E+31 1 41000 68 10 1 \n4 5 4.10E+31 1 41000 68 10 1 \n\n Sector state State\_Region ... pickle\_v sauce\_jam\_v Othrprocessed\_v \\\n0 2 24 242 ... 0.0 0.0 0.0 \n1 2 24 242 ... 0.0 0.0 0.0 \n2 2 24 242 ... 0.0 0.0 0.0 \n3 2 24 242 ... 0.0 0.0 0.0 \n4 2 24 242 ... 0.0 0.0 0.0 \n\n Beveragestotal\_v foodtotal\_v foodtotal\_q state\_1 Region \\\n0 0.000000 1141.492400 30.942394 GUJ 2 \n1 17.500000 1244.553500 29.286153 GUJ 2 \n2 0.000000 1050.315400 31.527046 GUJ 2 \n3 33.333333 1142.591667 27.834607 GUJ 2 \n4 75.000000 945.249500 27.600713 GUJ 2 \n\n fruits\_df\_tt\_v fv\_tot \n0 12.000000 154.18 \n1 333.000000 484.95 \n2 35.000000 214.84 \n3 168.333333 302.30 \n4 15.000000 148.00 \n\n[5 rows x 384 columns]\nIndex(['slno', 'grp', 'Round\_Centre', 'FSU\_number', 'Round', 'Schedule\_Number',\n 'Sample', 'Sector', 'state', 'State\_Region',\n ...\n 'pickle\_v', 'sauce\_jam\_v', 'Othrprocessed\_v', 'Beveragestotal\_v',\n 'foodtotal\_v', 'foodtotal\_q', 'state\_1', 'Region', 'fruits\_df\_tt\_v',\n 'fv\_tot'],\n dtype='object', length=384)\n","name":"stdout"}]},{"metadata":{"trusted":true},"id":"e78a0ea4-6740-4b40-91d1-1faf6af0f3e1","cell\_type":"code","source":"# Create a binary indicator for non-vegetarian status\ndata['non\_veg'] = ((data['nonvegtotal\_q'] > 0) | \n (data['eggsno\_q'] > 0) | \n (data['fishprawn\_q'] > 0) | \n (data['goatmeat\_q'] > 0) | \n (data['beef\_q'] > 0) | \n (data['pork\_q'] > 0) | \n (data['chicken\_q'] > 0) | \n (data['othrbirds\_q'] > 0)).astype(int)","execution\_count":4,"outputs":[]},{"metadata":{"trusted":true},"id":"c51b6ae7-f811-4e84-a9cc-a9d246b3f237","cell\_type":"code","source":"# Ensure the columns 'Age', 'MPCE\_URP', and 'Education' are present\nprint(data[['Age', 'MPCE\_URP', 'Education']].head())","execution\_count":5,"outputs":[{"output\_type":"stream","text":" Age MPCE\_URP Education\n0 50 3304.80 8.0\n1 40 7613.00 12.0\n2 45 3461.40 7.0\n3 75 3339.00 6.0\n4 30 2604.25 7.0\n","name":"stdout"}]},{"metadata":{"trusted":true},"id":"2971cb28-e15f-4a03-8e94-28373ae6acd1","cell\_type":"code","source":"# Drop rows with missing values in these columns\ndata.dropna(subset=['Age', 'MPCE\_URP', 'Education', 'non\_veg'], inplace=True)","execution\_count":6,"outputs":[]},{"metadata":{"trusted":true},"id":"fba4ec8a-f15a-4e14-b953-12402eaa0c61","cell\_type":"code","source":"# Prepare the data for the model\nX = data[['Age', 'MPCE\_URP', 'Education']]\ny = data['non\_veg']","execution\_count":7,"outputs":[]},{"metadata":{"trusted":true},"id":"2c6b69ba-556c-404a-8dd2-ad68935ce208","cell\_type":"code","source":"# Add a constant to the model (intercept)\nX = sm.add\_constant(X)","execution\_count":8,"outputs":[]},{"metadata":{"trusted":true},"id":"152a5ce8-054d-47ac-b351-ed1876b2d320","cell\_type":"code","source":"# Fit a probit regression model\nprobit\_model = sm.Probit(y, X).fit()\n\n# Summary of the model\nprint(probit\_model.summary())","execution\_count":9,"outputs":[{"output\_type":"stream","text":"Optimization terminated successfully.\n Current function value: 0.629775\n Iterations 4\n Probit Regression Results \n==============================================================================\nDep. Variable: non\_veg No. Observations: 101655\nModel: Probit Df Residuals: 101651\nMethod: MLE Df Model: 3\nDate: Mon, 01 Jul 2024 Pseudo R-squ.: 0.001666\nTime: 22:55:49 Log-Likelihood: -64020.\nconverged: True LL-Null: -64127.\nCovariance Type: nonrobust LLR p-value: 4.613e-46\n==============================================================================\n coef std err z P>|z| [0.025 0.975]\n------------------------------------------------------------------------------\nconst 0.5686 0.017 32.573 0.000 0.534 0.603\nAge -0.0002 0.000 -0.749 0.454 -0.001 0.000\nMPCE\_URP -2.932e-06 8.99e-07 -3.259 0.001 -4.69e-06 -1.17e-06\nEducation -0.0154 0.001 -13.467 0.000 -0.018 -0.013\n==============================================================================\n","name":"stdout"}]},{"metadata":{"trusted":true},"id":"bab213a8-56cf-4afc-bdd6-052c9a502b17","cell\_type":"code","source":"","execution\_count":null,"outputs":[]},{"metadata":{"trusted":true},"cell\_type":"code","source":"","execution\_count":null,"outputs":[]},{"metadata":{"trusted":true},"cell\_type":"code","source":"","execution\_count":null,"outputs":[]},{"metadata":{"trusted":true},"cell\_type":"code","source":"","execution\_count":null,"outputs":[]}]

**PART C**

### Interpretation of the Tobit Model Results

The Tobit model is an extension of the linear regression model designed to handle censoring. In this case, the binary indicator for non-vegetarian status (non\_veg) has been used with a left censoring at 0 and right censoring at 1, reflecting the binary nature of the variable.

**A screenshot of a computer

Description automatically generated**

### nterpretation of the Tobit Model Results

The Tobit model results provide insights into the relationship between the predictor variables (Age, MPCE\_URP, and Education) and the dependent variable non\_veg (non-vegetarian status). The model is designed to handle censoring at 0 and 1, reflecting the binary nature of the dependent variable.

#### Model Summary

* **Dependent Variable:** non\_veg
* **Number of Observations:** 101655
* **Model:** Tobit
* **Method:** Maximum Likelihood Estimation (MLE)
* **Log-Likelihood:** -73071
* **AIC (Akaike Information Criterion):** 1.462e+05
* **BIC (Bayesian Information Criterion):** 1.462e+05

#### Coefficients and Significance

1. **Intercept (const):**
   * **Coefficient:** 0.0052
   * **Standard Error:** 0.008
   * **z-value:** 0.692
   * **p-value:** 0.489
   * **95% Confidence Interval:** [-0.010, 0.020]
   * **Interpretation:** The intercept is not statistically significant, indicating that the baseline latent propensity for being non-vegetarian when all predictors are zero is not significantly different from zero.
2. **Age:**
   * **Coefficient:** 0.0107
   * **Standard Error:** 0.000
   * **z-value:** 82.978
   * **p-value:** 0.000 (highly significant)
   * **95% Confidence Interval:** [0.010, 0.011]
   * **Interpretation:** Age has a highly significant positive impact on the latent propensity to be non-vegetarian. As age increases, the likelihood of being non-vegetarian increases.
3. **MPCE\_URP (Monthly Per Capita Expenditure):**
   * **Coefficient:** -1.156e-06
   * **Standard Error:** 3.76e-07
   * **z-value:** -3.075
   * **p-value:** 0.002 (significant)
   * **95% Confidence Interval:** [-1.89e-06, -4.19e-07]
   * **Interpretation:** MPCE\_URP has a statistically significant negative impact on the latent propensity to be non-vegetarian. Higher monthly per capita expenditure is associated with a slightly lower likelihood of being non-vegetarian.
4. **Education:**
   * **Coefficient:** 0.0210
   * **Standard Error:** 0.000
   * **z-value:** 46.020
   * **p-value:** 0.000 (highly significant)
   * **95% Confidence Interval:** [0.020, 0.022]
   * **Interpretation:** Education has a highly significant positive impact on the latent propensity to be non-vegetarian. Higher levels of education are associated with an increased likelihood of being non-vegetarian.
5. **Sigma (par0):**
   * **Coefficient:** 0.4964
   * **Standard Error:** 0.001
   * **z-value:** 397.637
   * **p-value:** 0.000 (highly significant)
   * **95% Confidence Interval:** [0.494, 0.499]
   * **Interpretation:** Sigma represents the standard deviation of the error term. This value indicates moderate variability in the latent propensity to be non-vegetarian that is not explained by the predictors.

### Key Insights

1. **Age:** Age is a highly significant predictor, with a positive coefficient indicating that older individuals are more likely to be non-vegetarian.
2. **MPCE\_URP:** Monthly per capita expenditure has a small but statistically significant negative impact on the likelihood of being non-vegetarian. This suggests that individuals with higher expenditure are slightly less likely to be non-vegetarian.
3. **Education:** Education is also a highly significant predictor, with a positive coefficient indicating that higher education levels are associated with an increased likelihood of being non-vegetarian.
4. **Model Fit:** The Log-Likelihood value (-73071) and the information criteria (AIC and BIC) provide measures of the model's fit to the data. Lower values generally indicate a better fit, but they are most useful when comparing multiple models.

### Conclusion

The Tobit model has revealed significant relationships between the predictors (age, MPCE\_URP, and education) and the latent propensity to be non-vegetarian. Age and education both positively influence the likelihood of being non-vegetarian, while higher monthly per capita expenditure slightly decreases this likelihood. The model fits the data well and provides valuable insights into the factors that influence dietary choices.

**CODES**

[{"metadata":{"trusted":true},"id":"eaebd738-ba20-434e-8b1a-b943e6248b4b","cell\_type":"code","source":"import pandas as pd\nimport statsmodels.api as sm\nfrom statsmodels.base.model import GenericLikelihoodModel\nimport numpy as np\nfrom scipy.stats import norm","execution\_count":1,"outputs":[]},{"metadata":{"trusted":true},"id":"23cf49a4-fbb5-4f9f-84fb-19c0bc2442c6","cell\_type":"code","source":"# Load the data with low\_memory=False to avoid dtype warning\ndata = pd.read\_csv(\"/Users/janybalashiva/Downloads/NSSO68.csv\", low\_memory=False)\n\n# View the first few rows and columns\nprint(data.head())\nprint(data.columns)","execution\_count":2,"outputs":[{"output\_type":"stream","text":" slno grp Round\_Centre FSU\_number Round Schedule\_Number Sample \\\n0 1 4.10E+31 1 41000 68 10 1 \n1 2 4.10E+31 1 41000 68 10 1 \n2 3 4.10E+31 1 41000 68 10 1 \n3 4 4.10E+31 1 41000 68 10 1 \n4 5 4.10E+31 1 41000 68 10 1 \n\n Sector state State\_Region ... pickle\_v sauce\_jam\_v Othrprocessed\_v \\\n0 2 24 242 ... 0.0 0.0 0.0 \n1 2 24 242 ... 0.0 0.0 0.0 \n2 2 24 242 ... 0.0 0.0 0.0 \n3 2 24 242 ... 0.0 0.0 0.0 \n4 2 24 242 ... 0.0 0.0 0.0 \n\n Beveragestotal\_v foodtotal\_v foodtotal\_q state\_1 Region \\\n0 0.000000 1141.492400 30.942394 GUJ 2 \n1 17.500000 1244.553500 29.286153 GUJ 2 \n2 0.000000 1050.315400 31.527046 GUJ 2 \n3 33.333333 1142.591667 27.834607 GUJ 2 \n4 75.000000 945.249500 27.600713 GUJ 2 \n\n fruits\_df\_tt\_v fv\_tot \n0 12.000000 154.18 \n1 333.000000 484.95 \n2 35.000000 214.84 \n3 168.333333 302.30 \n4 15.000000 148.00 \n\n[5 rows x 384 columns]\nIndex(['slno', 'grp', 'Round\_Centre', 'FSU\_number', 'Round', 'Schedule\_Number',\n 'Sample', 'Sector', 'state', 'State\_Region',\n ...\n 'pickle\_v', 'sauce\_jam\_v', 'Othrprocessed\_v', 'Beveragestotal\_v',\n 'foodtotal\_v', 'foodtotal\_q', 'state\_1', 'Region', 'fruits\_df\_tt\_v',\n 'fv\_tot'],\n dtype='object', length=384)\n","name":"stdout"}]},{"metadata":{"trusted":true},"id":"8c10ac76-751a-480c-88b8-2296fa8725b2","cell\_type":"code","source":"# Create a binary indicator for non-vegetarian status\ndata['non\_veg'] = ((data['nonvegtotal\_q'] > 0) | \n (data['eggsno\_q'] > 0) | \n (data['fishprawn\_q'] > 0) | \n (data['goatmeat\_q'] > 0) | \n (data['beef\_q'] > 0) | \n (data['pork\_q'] > 0) | \n (data['chicken\_q'] > 0) | \n (data['othrbirds\_q'] > 0)).astype(int)","execution\_count":3,"outputs":[]},{"metadata":{"trusted":true},"id":"2788e1e8-4fe6-4c1d-948e-b3d9993914f2","cell\_type":"code","source":"# Ensure the columns 'Age', 'MPCE\_URP', and 'Education' are present\nprint(data[['Age', 'MPCE\_URP', 'Education']].head())","execution\_count":4,"outputs":[{"output\_type":"stream","text":" Age MPCE\_URP Education\n0 50 3304.80 8.0\n1 40 7613.00 12.0\n2 45 3461.40 7.0\n3 75 3339.00 6.0\n4 30 2604.25 7.0\n","name":"stdout"}]},{"metadata":{"trusted":true},"id":"0b80fbd0-610f-436b-b79c-cc06398602e5","cell\_type":"code","source":"# Drop rows with missing values in these columns\ndata.dropna(subset=['Age', 'MPCE\_URP', 'Education', 'non\_veg'], inplace=True)","execution\_count":5,"outputs":[]},{"metadata":{"trusted":true},"id":"b820b1a4-a0db-4ae9-b677-6f79867514c3","cell\_type":"code","source":"# Prepare the data for the model\nX = data[['Age', 'MPCE\_URP', 'Education']]\ny = data['non\_veg']\n\n# Add a constant to the model (intercept)\nX = sm.add\_constant(X)","execution\_count":6,"outputs":[]},{"metadata":{"trusted":true},"id":"0e026830-9510-4e08-8194-300e95946cc6","cell\_type":"code","source":"# Custom Tobit model\nclass Tobit(GenericLikelihoodModel):\n def \_\_init\_\_(self, endog, exog, left=None, right=None, \*\*kwds):\n self.left = left\n self.right = right\n super(Tobit, self).\_\_init\_\_(endog, exog, \*\*kwds)\n\n def nloglikeobs(self, params):\n exog = self.exog\n endog = self.endog\n beta = params[:-1]\n sigma = params[-1]\n xb = np.dot(exog, beta)\n z\_left = (self.left - xb) / sigma if self.left is not None else None\n z\_right = (self.right - xb) / sigma if self.right is not None else None\n\n ll = np.where(endog < self.left, np.log(norm.cdf(z\_left)),\n np.where(endog > self.right, np.log(norm.sf(z\_right)),\n norm.logpdf((endog - xb) / sigma) - np.log(sigma)))\n return -ll\n\n def fit(self, start\_params=None, maxiter=10000, maxfun=5000, \*\*kwds):\n if start\_params is None:\n start\_params = np.append(np.zeros(self.exog.shape[1]), 1)\n return super(Tobit, self).fit(start\_params=start\_params, maxiter=maxiter, maxfun=maxfun, \*\*kwds)\n\n# Fit the Tobit model\ntobit\_model = Tobit(y, X, left=0, right=1).fit()\n\n# Summary of the model\nprint(tobit\_model.summary())","execution\_count":7,"outputs":[{"output\_type":"stream","text":"/var/folders/z6/lcxc0qg53jdc\_26yqxr8f60c0000gn/T/ipykernel\_81973/223527810.py:17: RuntimeWarning: divide by zero encountered in log\n ll = np.where(endog < self.left, np.log(norm.cdf(z\_left)),\n/var/folders/z6/lcxc0qg53jdc\_26yqxr8f60c0000gn/T/ipykernel\_81973/223527810.py:18: RuntimeWarning: divide by zero encountered in log\n np.where(endog > self.right, np.log(norm.sf(z\_right)),\n","name":"stderr"},{"output\_type":"stream","text":"Optimization terminated successfully.\n Current function value: 0.718814\n Iterations: 212\n Function evaluations: 352\n Tobit Results \n==============================================================================\nDep. Variable: non\_veg Log-Likelihood: -73071.\nModel: Tobit AIC: 1.462e+05\nMethod: Maximum Likelihood BIC: 1.462e+05\nDate: Mon, 01 Jul 2024 \nTime: 23:05:37 \nNo. Observations: 101655 \nDf Residuals: 101651 \nDf Model: 3 \n==============================================================================\n coef std err z P>|z| [0.025 0.975]\n------------------------------------------------------------------------------\nconst 0.0052 0.008 0.692 0.489 -0.010 0.020\nAge 0.0107 0.000 82.978 0.000 0.010 0.011\nMPCE\_URP -1.156e-06 3.76e-07 -3.075 0.002 -1.89e-06 -4.19e-07\nEducation 0.0210 0.000 46.020 0.000 0.020 0.022\npar0 0.4964 0.001 397.637 0.000 0.494 0.499\n==============================================================================\n","name":"stdout"},{"output\_type":"stream","text":"/Users/janybalashiva/anaconda3/lib/python3.11/site-packages/statsmodels/base/model.py:2742: UserWarning: df\_model + k\_constant + k\_extra differs from k\_params\n warnings.warn(\"df\_model + k\_constant + k\_extra \"\n/Users/janybalashiva/anaconda3/lib/python3.11/site-packages/statsmodels/base/model.py:2746: UserWarning: df\_resid differs from nobs - k\_params\n warnings.warn(\"df\_resid differs from nobs - k\_params\")\n","name":"stderr"}]},{"metadata":{"trusted":false},"id":"bab06f48-1f0c-4d16-8281-33613dddf61c","cell\_type":"code","source":"","execution\_count":null,"outputs":[]}]