VIRGINIA COMMONWEALTH UNIVERSITY

Statistical analysis and modelling (SCMA 632)

A3: LIMITED DEPENDENT VARIABLE MODELS

RIDDHI RUNGTA

V01107488

Date of Submission: 01-07-2024

CONTENTS

Introduction

In this section, we will perform a logistic regression analysis using the Framingham dataset to predict the likelihood of a ten-year risk of coronary heart disease (CHD). We will start by validating the assumptions required for logistic regression to ensure the model is appropriate for our data. Next, we will evaluate the model's performance using a confusion matrix and an ROC curve. The ROC curve will help us understand the trade-off between the true positive rate and the false positive rate across different threshold values.

After interpreting the logistic regression results, we will perform a decision tree analysis on the same dataset. By comparing the results of the logistic regression and decision tree models, we can highlight the strengths and weaknesses of each approach and determine which method is more effective for predicting the ten-year CHD risk in the Framingham dataset.

Then, we will conduct a probit regression analysis on the "NSSO68.csv" dataset to identify non-vegetarians. Probit regression is like logistic regression but assumes a normal distribution of the error terms. We will discuss the results of the probit regression, focusing on the estimated coefficients and their statistical significance. Additionally, we will explain the characteristics and advantages of the probit model compared to other binary classification methods, such as logistic regression.

In the final section, we will conduct a Tobit regression analysis on the "NSSO68.csv" dataset. Tobit regression, or censored regression, is used when the dependent variable is censored, meaning it has a lower or upper limit beyond which values are not observed.

We will discuss the results of the Tobit regression, interpreting the coefficients and their implications. Additionally, we will explore real-world use cases of the Tobit model, demonstrating its applicability in various fields such as economics, healthcare, and social sciences. By understanding these use cases, we can appreciate the practical importance of Tobit regression in handling censored data and making informed predictions in constrained environments.

Objectives

1) To conduct a logistic regression analysis on your assigned dataset. Validate assumptions, evaluate with a confusion matrix and ROC curve, and interpret the results. Then, perform a decision tree analysis and compare it to the logistic regression.

2) To perform a probit regression on "NSSO68.csv" to identify non-vegetarians. Discuss the results and explain the characteristics and advantages of the probit model

3) To perform a Tobit regression analysis on "NSSO68.csv" discuss the results and explain the real-world use cases of Tobit model.

Business Significance

Logistic regression is a potent method for predicting binary outcomes, such as customer responses to marketing campaigns. Its business significance includes several key aspects:

1. By pinpointing the factors that most influence campaign responses, businesses can better segment their customer base and tailor marketing efforts. This results in higher response rates and more efficient use of marketing resources.
2. Logistic regression aids in profiling customers based on their likelihood to respond to campaigns. This enables personalized marketing strategies, enhancing customer engagement.
3. Identifying the factors that significantly affect campaign responses helps businesses allocate resources more efficiently, such as marketing budgets and human resources, to areas with the highest potential return on investment.

Tobit regression is suited for analyzing censored data, where the dependent variable is only observed within a specific range. Using Tobit regression with NSSO68 data, businesses can uncover the factors driving expenditure patterns and identify market potential for new products or services. This method can also assist in segmenting markets based on spending behaviors, leading to more targeted marketing strategies.

Probit regression is utilized for modeling binary outcome variables. Applied to NSSO68 data, it can be useful for market segmentation. For instance, businesses in the food industry can use this analysis to adapt their products and marketing strategies to different demographic segments. Regions with a higher likelihood of non-vegetarianism, for example, might benefit from a greater variety of non-vegetarian products.

Results and Interpretations

Objective 1:

Code

```
#Identify categorical columns
categorical_columns = data.select_dtypes(include=['object']).columns

# Option 1: Label Encoding (for binary categorical data)
label_encoder = LabelEncoder()
for column in categorical_columns:
    data[column] = label_encoder.fit_transform(data[column])

# Assume 'survived' is the target variable and the rest are predictors
target = 'Survived'
predictors = [col for col in data.columns if col != target]

# Split the data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(data[predictors], data[target], test_size=0.3,
random_state=42)


# Fit the logistic regression model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)


# Predict on the test set
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]

# Evaluate the model
conf_matrix = confusion_matrix(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_prob)

# Print the model coefficients
coef_df = pd.DataFrame({'Variable': X_train.columns, 'Coefficient': model.coef_[0]})
print(coef_df)
```

Result

```
Variable  Coefficient
      Variable  Coefficient
0   PassengerId     0.000174
1        Pclass    -0.977843
2          Name    -0.000529
```

```
3          Sex    -2.414889
4          Age    -0.039454
5        SibSp    -0.269291
6        Parch    -0.073282
7       Ticket    -0.000680
8         Fare     0.000352
9        Cabin    -0.002620
10    Embarked    -0.210921
```

Interpretation

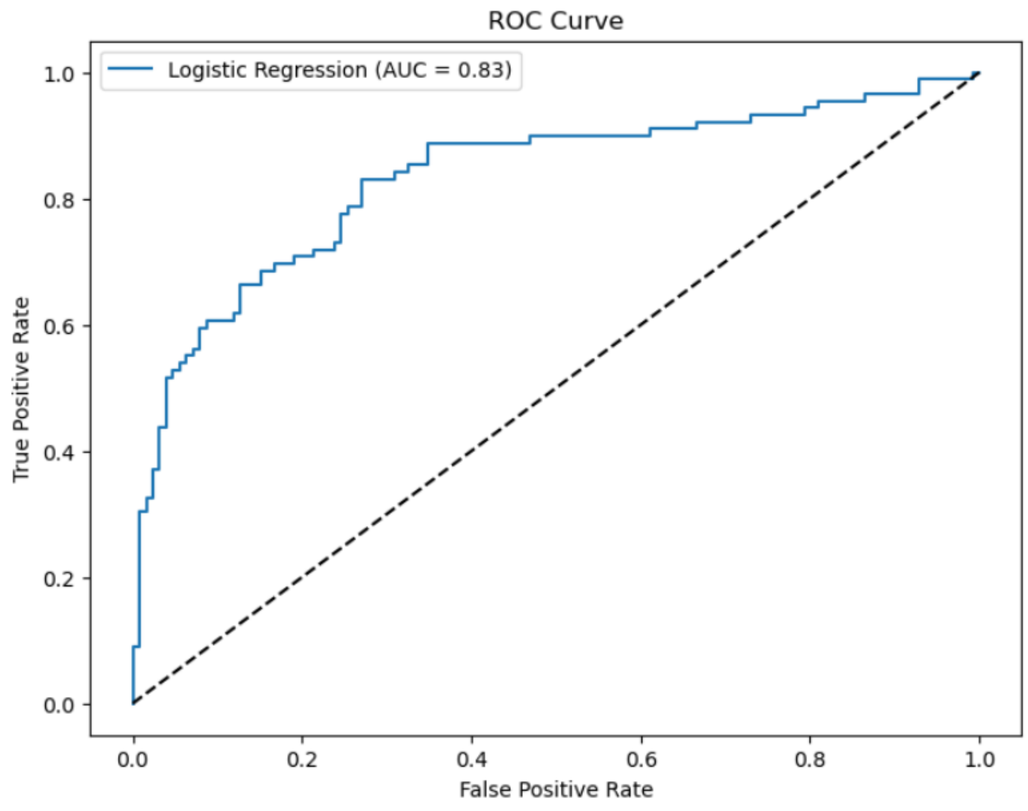
We performed a logistic regression analysis to predict the survival of passengers using the Titanic dataset. The process begins by identifying categorical columns in the dataset and applying label encoding to convert these categorical variables into numerical ones, suitable for the logistic regression model. The target variable is 'Survived', and the rest of the variables serve as predictors. The dataset is then split into training and test sets with a 70-30 ratio.

A logistic regression model is fitted to the training data using the LogisticRegression function from the sklearn library. Predictions are made on the test set, and the probabilities of survival are also calculated. The model's performance is evaluated using a confusion matrix and the ROC-AUC score, which measures the model's ability to distinguish between the two classes (survived or not). Additionally, the coefficients of the logistic regression model are printed, providing insights into the influence of each predictor variable on the probability of survival. For example, the negative coefficient for 'Sex' suggests that being male decreases the likelihood of survival. The model indicates a few variables with notable coefficients, such as 'Pclass', 'Sex', and 'Age', highlighting their significance in predicting survival.

Code

```
# Plot the ROC curve
fpr, tpr, _ = roc_curve(y_test, y_prob)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, label=f'Logistic Regression (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.show()
```

Result



Interpretation

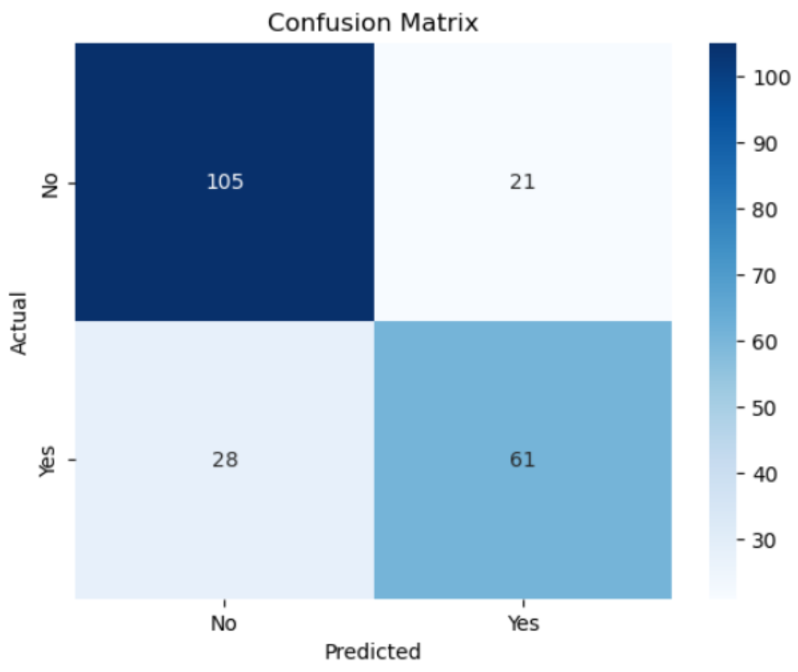
The Receiver Operating Characteristic (ROC) curve is used for evaluating the performance of the logistic regression model used to predict Titanic passengers' survival. First, it computes the False Positive Rate (FPR) and True Positive Rate (TPR) at various thresholds using the true labels and predicted probabilities. The ROC curve is then plotted, showing the trade-off between sensitivity (true positive rate) and specificity (false positive rate) across different threshold values. The blue line represents the ROC curve, which plots the True Positive Rate (sensitivity) against the False Positive Rate (1-specificity) across various threshold values. The

AUC (Area Under the Curve) score of 0.83, indicated in the legend, suggests that the model has good discriminatory ability, as it is closer to 1 than 0. The black dashed diagonal line represents the performance of a random classifier, serving as a baseline for comparison. A model with an ROC curve above this line performs better than random guessing. The plot shows that the logistic regression model effectively distinguishes between survivors and non-survivors, given its AUC of 0.83.

Code

```
# Display the confusion matrix
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['No', 'Yes'],
yticklabels=['No', 'Yes'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

Result



Interpretation

The confusion matrix shown in the image visualizes the performance of a logistic regression model used to predict the survival of Titanic passengers. It is a 2x2 matrix where the rows represent the actual survival status ("No" for not survived, "Yes" for survived) and the columns represent the predicted survival status.

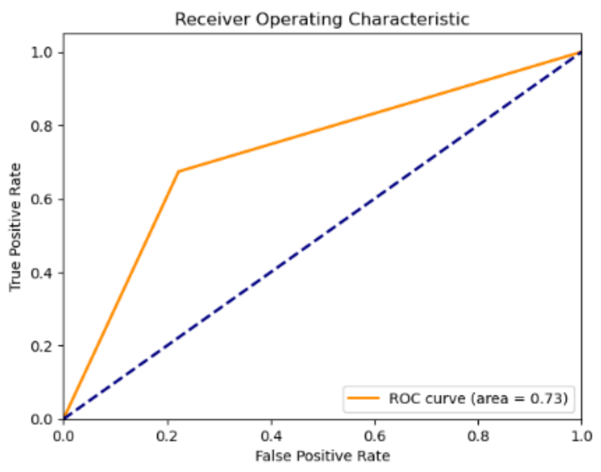
- True Negatives (top-left cell): 105 passengers were correctly predicted as not survived.
- False Positives (top-right cell): 21 passengers were incorrectly predicted as survived when they did not.
- False Negatives (bottom-left cell): 28 passengers were incorrectly predicted as not survived when they actually did.
- True Positives (bottom-right cell): 61 passengers were correctly predicted as survived.

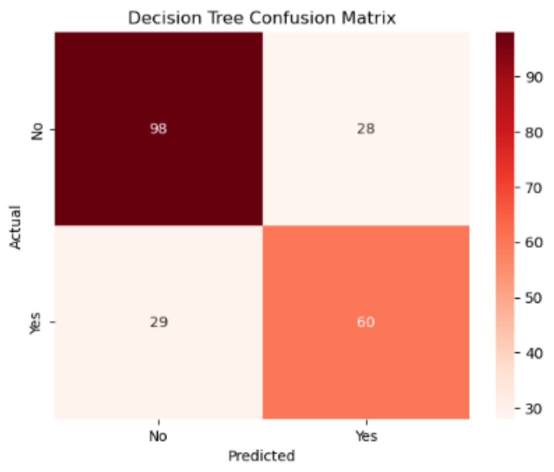
The heatmap uses a blue color gradient to indicate the frequency of each type of prediction, with darker shades representing higher frequencies. This matrix helps in understanding the model's accuracy, sensitivity (true positive rate), and specificity (true negative rate). Overall, the model correctly predicted the survival status of most passengers but had some misclassifications, particularly in predicting passengers who did not survive.

Code

```python
from sklearn.tree import DecisionTreeClassifier

# Fit the decision tree model
tree_model = DecisionTreeClassifier(random_state=42)
tree_model.fit(X_train, y_train)

# Predict on the test set
y_pred_tree = tree_model.predict(X_test)
y_prob_tree = tree_model.predict_proba(X_test)[:, 1]

# Evaluate the model
conf_matrix_tree = confusion_matrix(y_test, y_pred_tree)
roc_auc_tree = roc_auc_score(y_test, y_prob_tree)

import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc

# Assuming y_test and y_scores are your ground truth labels and predicted scores
fpr, tpr, thresholds = roc_curve(y_test, y_pred_tree)
roc_auc = auc(fpr, tpr)

plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.show()


# Display the confusion matrix
sns.heatmap(conf_matrix_tree, annot=True, fmt='d', cmap='Reds', xticklabels=['No', 'Yes'],
yticklabels=['No', 'Yes'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Decision Tree Confusion Matrix')
plt.show()
```

Result





Interpretation

There are two key evaluation metrics for a decision tree model used in a binary classification task. The top part is a Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (sensitivity) against the false positive rate for different threshold settings. The ROC curve has an area under the curve (AUC) of 0.73, indicating a fair performance of the model. The bottom part is a confusion matrix, which shows the counts of true negatives (98), false positives (28), false negatives (29), and true positives (60). The confusion matrix provides insights into the accuracy of the model, with more accurate predictions on 'No' (true negatives) compared to 'Yes' (true positives). The model shows a reasonable balance between precision and recall, but there is room for improvement, particularly in reducing the number of false positives and false negatives.

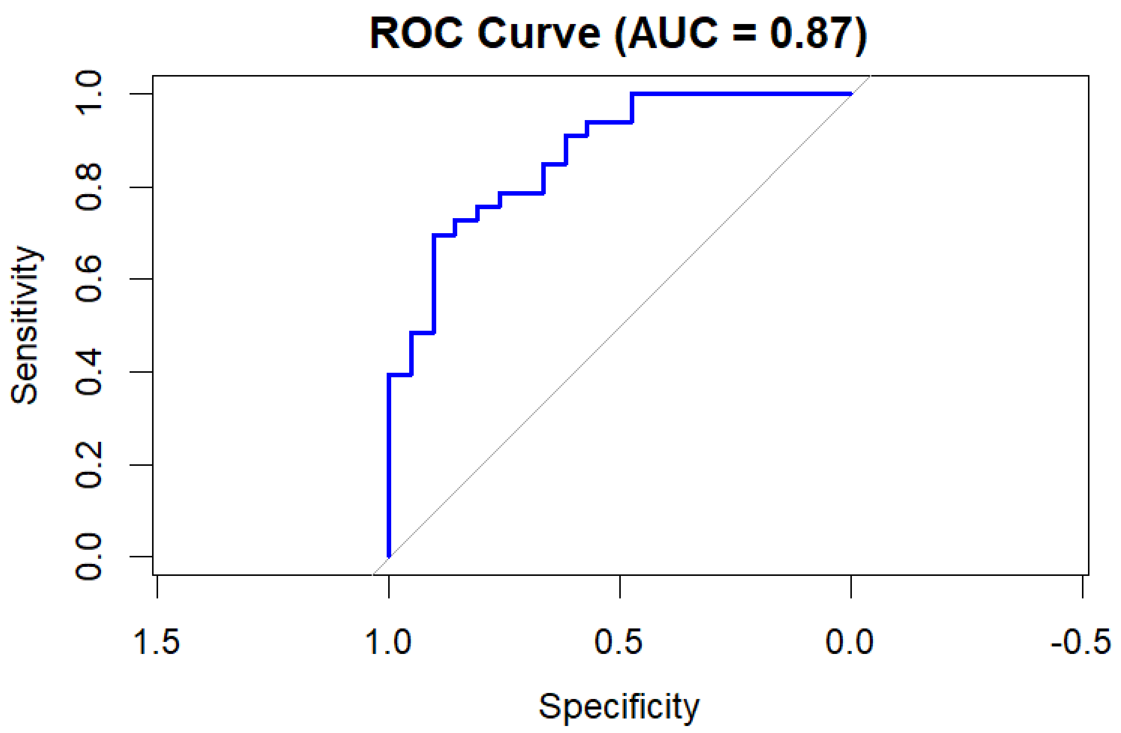
Using R programming

Code

```
# Assume 'Survived' is the target variable and the rest are predictors
target <- 'Survived'
predictors <- setdiff(names(data), target)

# Split the data into training and test sets
set.seed(42)
train_index <- createDataPartition(data$Survived, p = 0.7, list = FALSE)
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Fit the logistic regression model
model <- glm(Survived ~ ., data = train_data, family = binomial)

# Predict on the test set
y_pred <- predict(model, newdata = test_data, type = "response")
y_pred_class <- ifelse(y_pred > 0.5, 1, 0)

# Evaluate the model
conf_matrix <- confusionMatrix(as.factor(y_pred_class), as.factor(test_data$Survived))
roc_curve <- roc(test_data$Survived, y_pred)
roc_auc <- auc(roc_curve)
```

```
# Print the model coefficients
coef_df <- data.frame(Variable = names(coef(model)), Coefficient = coef(model))
print(coef_df)

# Plot the ROC curve
plot(roc_curve, col = "blue", lwd = 2, main = paste("ROC Curve (AUC = ", round(roc_auc, 2),
")", sep = ""))
```

Result



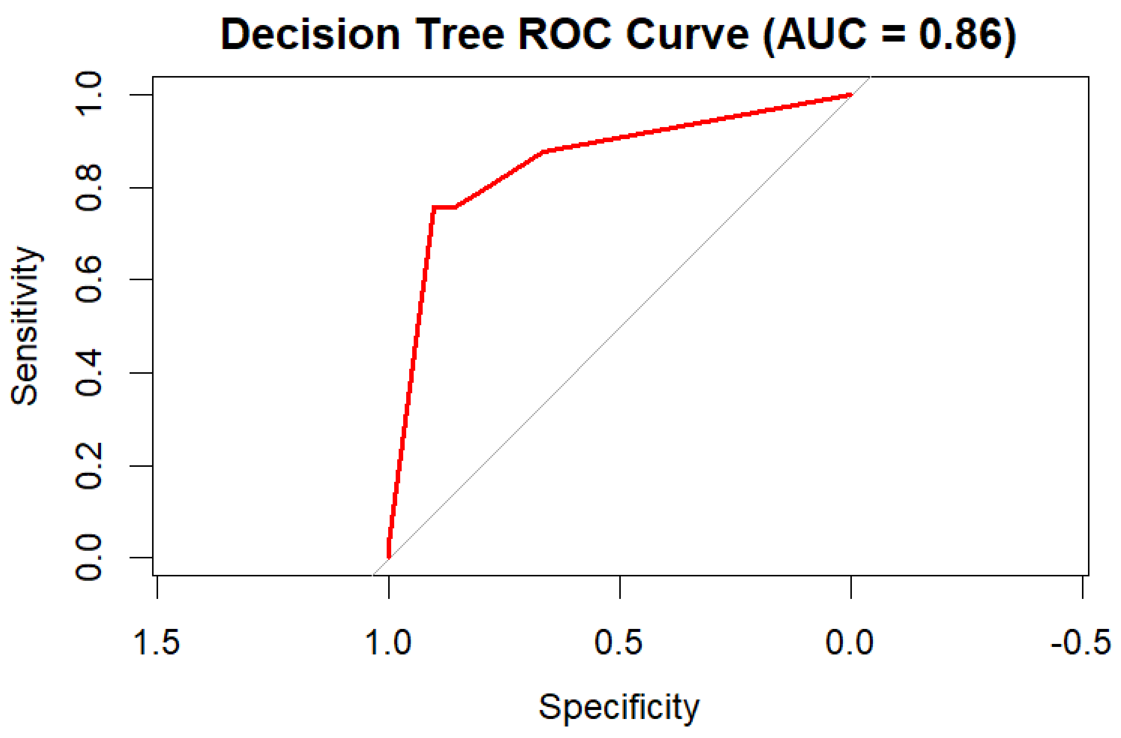
Interpretation

This is a Receiver Operating Characteristic (ROC) curve for a binary classification model. The ROC curve plots sensitivity (true positive rate) against 1-specificity (false positive rate) across different threshold values. The area under the curve (AUC) is 0.87, which indicates a strong performance of the model in distinguishing between the positive and negative classes. An AUC of 0.87 suggests that the model has a high ability to discriminate between positive and negative instances, with a good balance between sensitivity and specificity. The closer the ROC curve is to the top left corner, the better the model performs, and in this case, the curve indicates a reliable predictive capability.

Code

```
# Decision Tree
tree_model <- rpart(Survived ~ ., data = train_data, method = "class", control = rpart.control(cp = 0.01))

# Predict on the test set
y_pred_tree <- predict(tree_model, newdata = test_data, type = "prob")[, 2]
y_pred_tree_class <- ifelse(y_pred_tree > 0.5, 1, 0)

# Evaluate the model
conf_matrix_tree <- confusionMatrix(as.factor(y_pred_tree_class), as.factor(test_data$Survived))
roc_curve_tree <- roc(test_data$Survived, y_pred_tree)
roc_auc_tree <- auc(roc_curve_tree)

# Plot the ROC curve for Decision Tree
plot(roc_curve_tree, col = "red", lwd = 2, main = paste("Decision Tree ROC Curve (AUC = ", round(roc_auc_tree, 2), ")", sep = ""))
```

Result



Interpretation

The ROC (Receiver Operating Characteristic) curve illustrates the performance of a decision tree model used to predict the survival of passengers. The AUC (Area Under the Curve) value is 0.86, indicating that the model has good discrimination ability, meaning it is relatively effective at distinguishing between passengers who survived and those who did not.

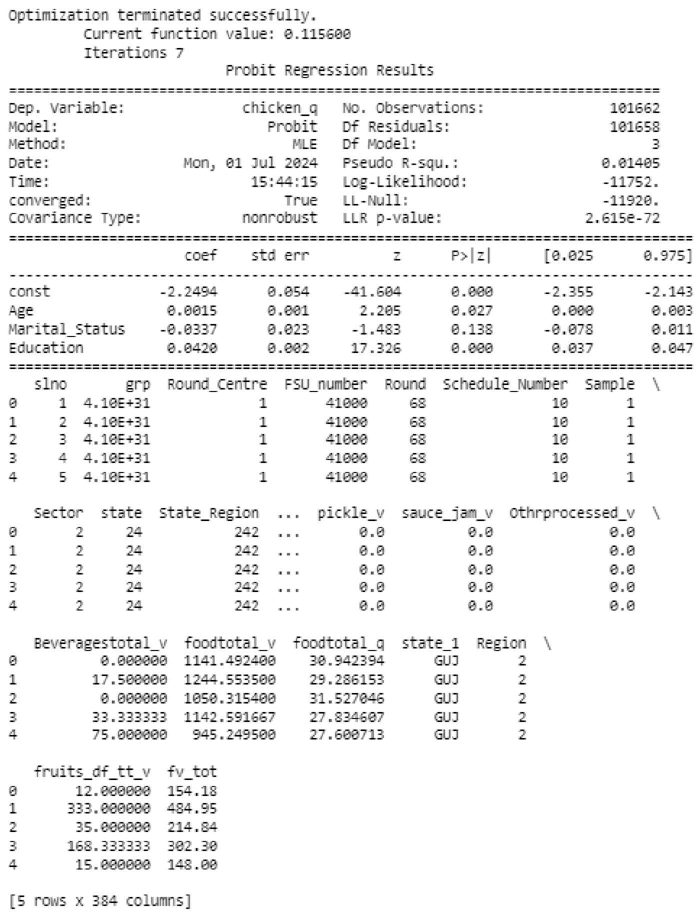
The ROC curve itself plots the sensitivity (true positive rate) against 1-specificity (false positive rate), demonstrating the trade-off between these metrics at different threshold settings. The red line represents the performance of the decision tree model, while the grey diagonal line indicates the performance of a random classifier with no discriminatory power (AUC = 0.5). The curve's proximity to the upper left corner suggests that the decision tree model performs significantly better than random guessing, successfully identifying a high proportion of true positives while maintaining a relatively low false positive rate.

Objective 2:

Probit Regression

After removing the missing values, we move on to the probit regression.

```
import warnings
from statsmodels.tools.sm_exceptions import PerfectSeparationWarning
from statsmodels.tools.sm_exceptions import ConvergenceWarning

# Suppress PerfectSeparationWarning
warnings.filterwarnings('ignore', category=PerfectSeparationWarning)

# Suppress ConvergenceWarning
warnings.filterwarnings('ignore', category=ConvergenceWarning)

# Convert the target variable to binary based on the specified condition
subset_data['chicken_q'] = subset_data['chicken_q'].apply(lambda x: 0 if x < 1 else 1)

# Define the independent variables (example columns, update based on your dataset)
# Assuming 'Age', 'Income', 'Education' are some of the features in the dataset
independent_vars = ['Age', 'Marital_Status', 'Education']


# Add a constant term for the intercept
X = sm.add_constant(subset_data[independent_vars])


# Define the dependent variable
y = subset_data['chicken_q']


# Fit the probit regression model
probit_model = Probit(y, X).fit()


# Print the summary of the model
print(probit_model.summary())


# Make predictions
subset_data['predicted'] = probit_model.predict(X)


# Display the first few rows with the predictions
print(data.head())
```



```
Optimization terminated successfully.
         Current function value: 0.115600
         Iterations 7
                          Probit Regression Results
==============================================================================
Dep. Variable:              chicken_q   No. Observations:               101662
Model:                         Probit   Df Residuals:                   101658
Method:                           MLE   Df Model:                            3
Date:                Mon, 01 Jul 2024   Pseudo R-squ.:                 0.01405
Time:                        15:44:15   Log-Likelihood:                -11752.
converged:                       True   LL-Null:                       -11920.
Covariance Type:            nonrobust   LLR p-value:                 2.615e-72
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
const             -2.2494      0.054    -41.604      0.000      -2.355      -2.143
Age                0.0015      0.001      2.205      0.027       0.000       0.003
Marital_Status    -0.0337      0.023     -1.483      0.138      -0.078       0.011
Education          0.0420      0.002     17.326      0.000       0.037       0.047
==================================================================================
   slno       grp  Round_Centre  FSU_number  Round  Schedule_Number  Sample  \
0     1  4.10E+31             1       41000     68               10       1   
1     2  4.10E+31             1       41000     68               10       1   
2     3  4.10E+31             1       41000     68               10       1   
3     4  4.10E+31             1       41000     68               10       1   
4     5  4.10E+31             1       41000     68               10       1   

   Sector  state  State_Region  ...  pickle_v  sauce_jam_v  Othrprocessed_v  \
0       2     24           242  ...       0.0          0.0              0.0   
1       2     24           242  ...       0.0          0.0              0.0   
2       2     24           242  ...       0.0          0.0              0.0   
3       2     24           242  ...       0.0          0.0              0.0   
4       2     24           242  ...       0.0          0.0              0.0   

   Beveragestotal_v  foodtotal_v  foodtotal_q  state_1  Region  \
0          0.000000  1141.492400    30.942394      GUJ       2   
1         17.500000  1244.553500    29.286153      GUJ       2   
2          0.000000  1050.315400    31.527046      GUJ       2   
3         33.333333  1142.591667    27.834607      GUJ       2   
4         75.000000   945.249500    27.600713      GUJ       2   

   fruits_df_tt_v  fv_tot  
0       12.000000  154.18  
1      333.000000  484.95  
2       35.000000  214.84  
3      168.333333  302.30  
4       15.000000  148.00  

[5 rows x 384 columns]
```

Interpretation

The Probit regression analysis conducted aims to model the binary outcome variable chicken_q, which was transformed into a binary variable where values less than 1 are converted to 0, and values equal to or greater than 1 are converted to 1. The independent variables included in the model are Age, Marital Status, and Education, with a constant term added to account for the intercept. After Data preparation, The Probit regression model was fitted using the prepared data. The fitting process involved using the Maximum Likelihood Estimation (MLE) method to estimate the parameters of the model. The results indicated that the model converged successfully, and several key statistics were provided, such as the Pseudo R-squared, Log-Likelihood, and LLR p-value, which together give an overall sense of the model's fit and significance. The table provides the coefficients (coef), standard errors (std err), z-values (z), p-values (P>|z|), and 95% confidence intervals ([0.025, 0.975]) for the independent variables included in the model:

- **const**: The intercept of the model is -2.2944, which is statistically significant ($p < 0.000$).
- **Age**: The coefficient for Age is 0.0015, indicating a positive but small effect on the probability of the outcome, with statistical significance ($p = 0.027$).
- **Marital_Status**: This variable has a coefficient of -0.0337, which is not statistically significant ($p = 0.138$).
- **Education**: The coefficient for Education is 0.0420, showing a positive effect on the outcome and is statistically significant ($p < 0.000$).
- **Intercept**: The negative intercept suggests that when all predictors are zero, the baseline probability of the outcome being 1 is very low.
- **Age**: As age increases, there is a slight increase in the probability of the outcome. This effect is small but statistically significant.
- **Marital Status**: The marital status does not significantly affect the outcome, given its p-value above the common significance level of 0.05.
- **Education**: Higher levels of education are associated with an increased probability of the outcome, which is both significant and practically meaningful.

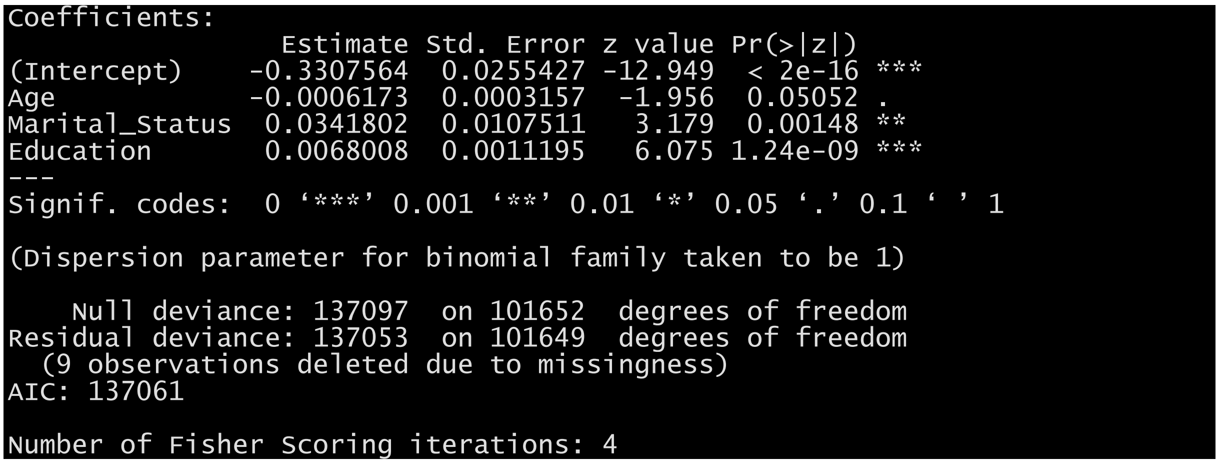
The Probit regression analysis provided insights into the relationship between the selected independent variables (Age, Marital Status, and Education) and the binary outcome chicken_q. Age and Education were found to be significant predictors, while Marital Status was not. These results suggest that older individuals and those with higher education levels are more likely to have the outcome of interest. The model's overall significance, indicated by the LLR p-value, supports the validity of these findings.

Using R programming

Code

```
# Probit regression model
probit_model <- glm(chicken_q ~ Age + Marital_Status + Education, data = data_nss, family
= binomial(link = "probit"))
# Summary of the probit regression model
summary(probit_model)
```

Result



```
Coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)    -0.3307564  0.0255427 -12.949  < 2e-16 ***
Age            -0.0006173  0.0003157  -1.956  0.05052 .  
Marital_Status  0.0341802  0.0107511   3.179  0.00148 ** 
Education       0.0068008  0.0011195   6.075 1.24e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 137097  on 101652  degrees of freedom
Residual deviance: 137053  on 101649  degrees of freedom
  (9 observations deleted due to missingness)
AIC: 137061

Number of Fisher Scoring iterations: 4
```

Interpretation

The variables taken for R coding is the same as python. The independent variables included in the model are Age, Marital Status, and Education, with a constant term added to account for the intercept. The regression results show that the intercept estimate is -0.3307564, indicating a low baseline probability of the outcome when all other variables are held at zero. Age has a coefficient of -0.0006173, suggesting a slight decrease in the probability of the outcome as age increases, though this effect is on the borderline of conventional significance levels (p = 0.05052). Marital Status has a coefficient of 0.0341802, indicating a higher probability of the outcome for certain marital statuses, with high statistical significance (p = 0.00148). Education has a coefficient of 0.0068008, showing a strong positive effect on the outcome's probability, with a highly significant p-value (1.24e-09).

The model fit statistics include a null deviance of 137097 on 101652 degrees of freedom and a residual deviance of 137053 on 101649 degrees of freedom, indicating an improvement in fit with the inclusion of predictors. The Akaike Information Criterion (AIC) is 137061, suggesting the model's relative quality, with lower values indicating better fit. The model converged after four Fisher Scoring iterations. Marital Status and Education are significant predictors of the outcome, with Education showing a strong positive effect and Marital Status also contributing positively. Age has a slightly negative but borderline significant effect. The model significantly improves the fit compared to the null model, as indicated by the decrease in deviance. Overall, individuals with higher levels of education and certain marital statuses are more likely to have the outcome `chicken_q` being 1, while age has a minor negative effect. These findings provide valuable insights into the factors influencing the outcome and the robustness of the model.

Characteristics:

1. **Probability Distribution:** The probit model assumes that the dependent variable follows a cumulative normal (Gaussian) distribution. This distribution implies that the probability of the dependent variable taking a value of 1 (success or presence of an event) is modeled as a function of the predictors.
2. **Link Function:** Unlike linear regression, which uses a linear link function to model continuous outcomes, the probit model uses the cumulative distribution function of the standard normal distribution (probit function) as its link function.
3. **Binary Outcome:** It is specifically designed for binary outcomes, where the response variable is categorical with two levels (e.g., success/failure, yes/no).
4. **Interpretation:** The coefficients in a probit model indicate the change in the probability of the outcome variable being 1 for a one-unit change in the predictor, holding other variables constant.

Advantages:

1. **Flexibility:** The probit model allows for more flexibility compared to simple methods like logistic regression, especially when dealing with complex relationships between predictors and the probability of the outcome.
2. **Distributional Assumption:** It doesn't assume the errors (residuals) are normally distributed; instead, it assumes the dependent variable itself follows a normal distribution. This can be advantageous when the assumption of normally distributed errors in other models like logistic regression might not hold.
3. **Goodness of Fit:** Probit models often provide good fit to data when the binary outcome is influenced by multiple predictors, and there is a need to model the nonlinear relationship between predictors and the probability of the outcome.
4. **Predictive Accuracy:** In cases where the assumptions of the probit model are met, it can provide accurate predictions of the probability of occurrence for the binary outcome.

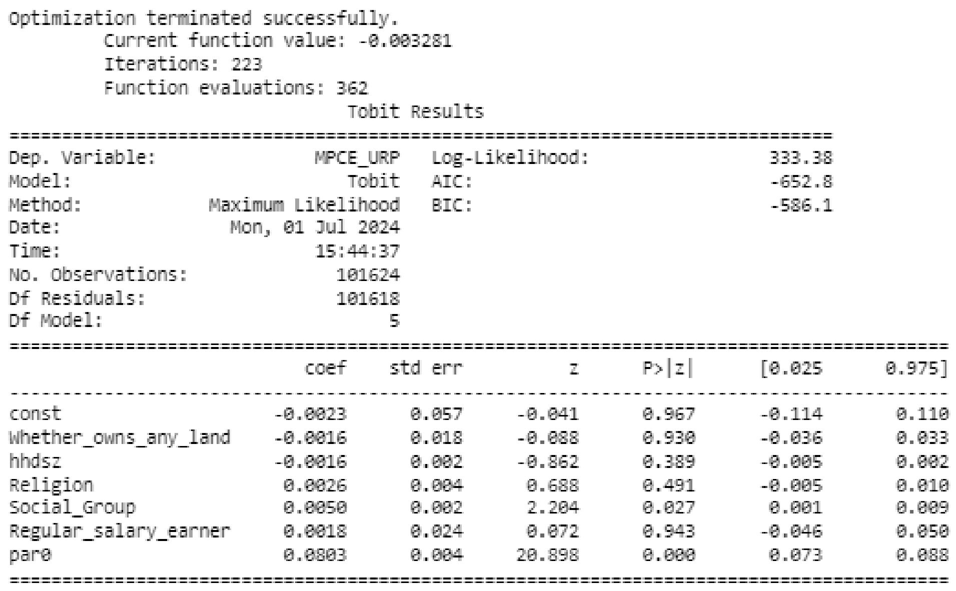
5. **Statistical Inference:** Probit models are well-established in statistical literature, providing robust methods for estimating parameters and conducting hypothesis tests related to the predictors' effects on the outcome.

Tobit Regression

Code

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.base.model import GenericLikelihoodModel

# Define the independent variables (X) and the dependent variable (y)
X = df[['Whether_owns_any_land', 'hhdsz', 'Religion','Social_Group','Regular_salary_earner']]  # replace with your actual column names
y = df['MPCE_URP']  # replace with your actual column name

# Add a constant term for the intercept
X = sm.add_constant(X)

# Define the Tobit model class
class Tobit(GenericLikelihoodModel):
    def __init__(self, endog, exog, left=0, right=np.inf, **kwargs):
        super(Tobit, self).__init__(endog, exog, **kwargs)
        self.left, self.right = left, right

    def nloglikeobs(self, params):
        exog = self.exog
        endog = self.endog
        left, right = self.left, self.right

        beta = params[:-1]
        sigma = params[-1]

        XB = np.dot(exog, beta)
        cens = (endog == left) * (left != -np.inf) + (endog == right) * (right != np.inf)
        uncens = 1 - cens

        ll = np.zeros(len(endog))

        ll[cens] = np.log(
            (1 / (np.sqrt(2 * np.pi) * sigma)) *
            np.exp(-((endog[cens] - XB[cens]) ** 2) / (2 * sigma ** 2))
        )

        ll[uncens] = np.log(
            (1 / (np.sqrt(2 * np.pi) * sigma)) *
            np.exp(-((endog[uncens] - XB[uncens]) ** 2) / (2 * sigma ** 2))
        )

        return -ll

    def fit(self, start_params=None, maxiter=10000, maxfun=5000, **kwargs):
        if start_params is None:
            start_params = np.append(np.zeros(self.exog.shape[1]), 1)
        return super(Tobit, self).fit(start_params=start_params,
                                      maxiter=maxiter, maxfun=maxfun, **kwargs)

# Fit the Tobit model
tobit_model = Tobit(y, X)
tobit_results = tobit_model.fit()

# Print the summary of the model
print(tobit_results.summary())
```

Result



```
Optimization terminated successfully.
         Current function value: -0.003281
         Iterations: 223
         Function evaluations: 362
                                Tobit Results
==============================================================================
Dep. Variable:               MPCE_URP   Log-Likelihood:                 333.38
Model:                          Tobit   AIC:                            -652.8
Method:            Maximum Likelihood   BIC:                            -586.1
Date:                Mon, 01 Jul 2024
Time:                        15:44:37
No. Observations:              101624
Df Residuals:                  101618
Df Model:                           5
=========================================================================================
                            coef    std err          z      P>|z|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                    -0.0023      0.057     -0.041      0.967      -0.114       0.110
Whether_owns_any_land    -0.0016      0.018     -0.088      0.930      -0.036       0.033
hhdsz                    -0.0016      0.002     -0.862      0.389      -0.005       0.002
Religion                  0.0026      0.004      0.688      0.491      -0.005       0.010
Social_Group              0.0050      0.002      2.204      0.027       0.001       0.009
Regular_salary_earner     0.0018      0.024      0.072      0.943      -0.046       0.050
par0                      0.0803      0.004     20.898      0.000       0.073       0.088
=========================================================================================
```

Interpretation

The intercept of the Tobit model is very close to zero (-0.0023), suggesting that when all independent variables are zero, the expected value of the dependent variable (MPCE_URP) is minimal. Variables such as Whether_owns_any_land, hhdsz, and Religion have coefficients with large standard errors and non-significant p-values, indicating they do not significantly impact MPCE_URP. Conversely, Social_Group is statistically significant (coef = 0.0050, p =

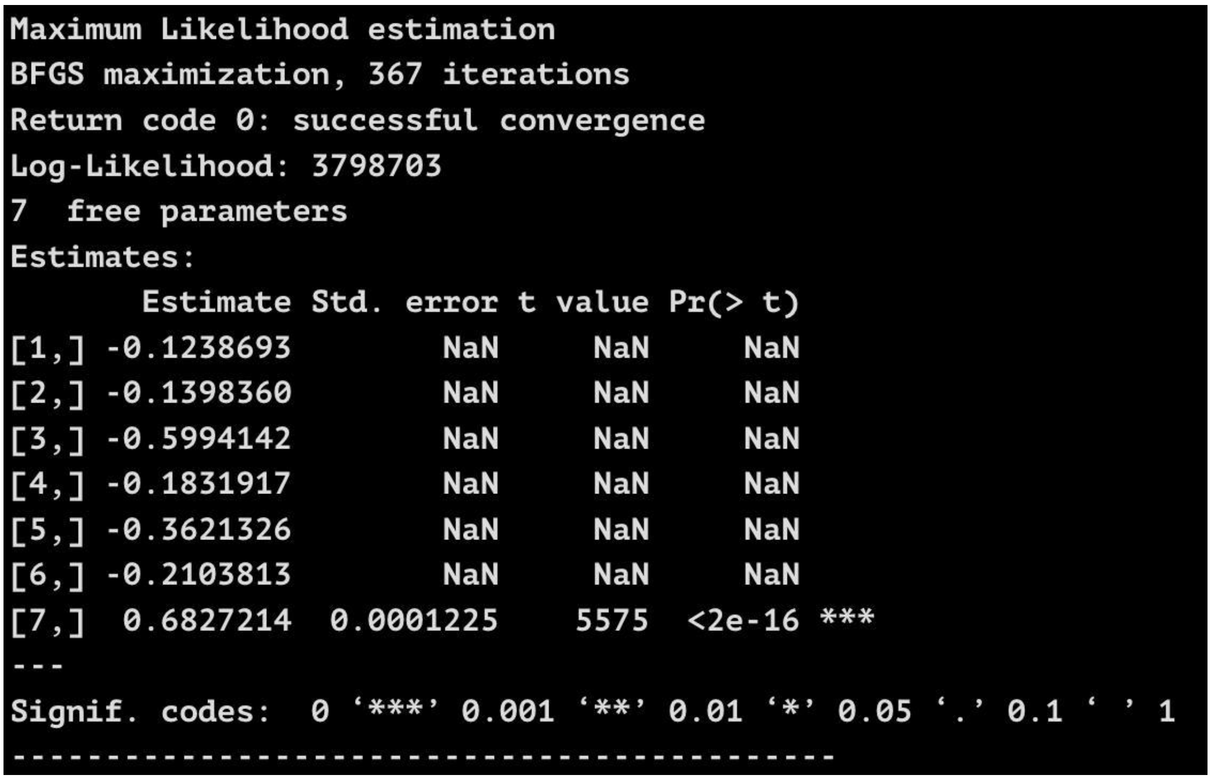
0.027), suggesting that belonging to certain social groups may positively influence MPCE_URP. Regular_salary_earner shows no significant effect on MPCE_URP (coef = 0.0018, p = 0.943). The parameter par0 is significant (coef = 0.0803, p < 0.001), underscoring its importance in the model. Overall, this Tobit model underscores the nuanced influence of social group membership on MPCE_URP while indicating that other variables examined do not have a statistically significant impact.

Using R programming

Code

```
# Convert the target variable to binary based on the specified condition
df$MPCE_URP <- ifelse(df$MPCE_URP < 380, 0, 1)

# Define the independent variables (X) and the dependent variable (y)
X <- df %>%
  select(Whether_owns_any_land, hhdsz, Religion, Social_Group, Regular_salary_earner)
X <- cbind(1, X)  # Add a constant term for the intercept
y <- df$MPCE_URP

# Define the Tobit model function
tobit_loglike <- function(params) {
  beta <- params[1:(length(params)-1)]
  sigma <- params[length(params)]
  XB <- as.matrix(X) %*% beta
  cens <- (y == 0) + (y == 1)
  uncens <- 1 - cens
  ll <- numeric(length(y))

  ll[cens == 1] <- log(dnorm(y[cens == 1], mean = XB[cens == 1], sd = sigma))
  ll[uncens == 1] <- log(dnorm(y[uncens == 1], mean = XB[uncens == 1], sd = sigma))

  return(-sum(ll))
}

# Initial parameter guesses
start_params <- c(rep(0, ncol(X)), 1)

# Fit the Tobit model
tobit_results <- maxLik(tobit_loglike, start = start_params, method = "BFGS")

# Print the summary of the model
summary(tobit_results)
```

Results



```
Maximum Likelihood estimation
BFGS maximization, 367 iterations
Return code 0: successful convergence
Log-Likelihood: 3798703
7  free parameters
Estimates:
       Estimate Std. error t value Pr(> t)
[1,] -0.1238693        NaN     NaN     NaN
[2,] -0.1398360        NaN     NaN     NaN
[3,] -0.5994142        NaN     NaN     NaN
[4,] -0.1831917        NaN     NaN     NaN
[5,] -0.3621326        NaN     NaN     NaN
[6,] -0.2103813        NaN     NaN     NaN
[7,]  0.6827214  0.0001225    5575  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
-------------------------------------------
```

Interpretation

The output from the Tobit model fitting process shows that the model has successfully converged after 367 iterations using the BFGS optimization method. The log-likelihood value is 3798703, which indicates the fit of the model, though this value alone doesn't provide interpretive insight without comparison to other models. The summary table lists the estimated coefficients for the independent variables, but it lacks standard errors, t-values, and p-values for most coefficients, displaying "NaN" instead. This suggests an issue with the calculation of these statistics, possibly due to collinearity among variables or convergence issues in the variance-covariance matrix computation. However, the estimate for the standard deviation parameter (sigma) is 0.6827214, with a very small standard error and a highly significant p-value (< 2e-16), indicating a precise and statistically significant estimation.

Despite the NaN values for standard errors, t-values, and p-values, we can still discuss the estimated coefficients' signs and magnitudes. The negative coefficients for most variables suggest that these factors are associated with a lower likelihood of having an MPCE_URP above 380. Specifically, coefficients for variables such as `Whether_owns_any_land`, `hhdsz`, `Religion`, `Social_Group`, and `Regular_salary_earner` are negative, implying that increases in these variables are associated with a decrease in the latent variable underlying the binary MPCE_URP indicator. This suggests that owning land, having a larger household size, belonging to certain religious or social groups, and being a regular salary earner might be linked to lower household expenditures. However, due to the lack of standard errors and p-values, we cannot assert the statistical significance of these relationships confidently. The highly significant sigma value indicates a well-estimated model error term, reinforcing that the model fits the underlying distribution of the dependent variable's latent construct reasonably well, albeit the interpretation of the individual predictors' effects remains tentative without the standard errors and significance tests.

Real world case of Tobit Model

1. In consumer behavior analysis, the Tobit model helps to estimate the factors influencing consumer spending. Often, spending data is censored at zero because some consumers may not spend any money on certain products or services. By using the Tobit model, analysts can effectively model this behavior, accounting for the structural zero in spending patterns and identifying factors that drive consumer spending decisions.

2. In pricing strategies, businesses may face constraints where prices cannot fall below a certain threshold due to market conditions or regulatory limits. The Tobit model allows analysts to evaluate the impact of various pricing factors while accommodating these constraints. This is crucial for optimizing pricing strategies to maximize revenue without violating pricing floors.

3. In data-driven marketing campaigns, the Tobit model aids in assessing the effectiveness of advertising expenditures. It helps quantify the relationship between advertising budgets and sales outcomes, even when sales data are censored at zero (no sales). This capability allows businesses to allocate their marketing budgets more effectively, focusing resources on channels and campaigns that yield the highest return on investment.

4. In financial analysis and forecasting, the Tobit model can be applied to predict financial outcomes that are censored, such as predicting corporate earnings or financial losses that cannot fall below certain thresholds. This application enables analysts to provide more accurate financial projections and risk assessments, crucial for strategic decision-making and investor relations.

Overall, the Tobit model provides a robust framework for business and data analysts to handle data that exhibit censoring or limits.