VIRGINIA COMMONWEALTH UNIVERSITY

STATISTICAL ANALYSIS & MODELING

A1b: INDIAN PREMIER LEAGUE PLAYER DATA ANALYSIS USING PYTHON AND R

Jany Balasivan
V01108262

Date of Submission: 23/06/2024

CONTENTS

Introduction

Loan Eligibility Prediction Analysis

Loan eligibility prediction is a critical task for financial institutions to determine the likelihood of an applicant repaying a loan. Using historical data and machine learning techniques, we can build predictive models to assist in making informed lending decisions. This documentation provides an analysis of a loan eligibility dataset, focusing on logistic regression and decision tree models to predict loan eligibility.

Objectives

1. Conduct a Logistic Regression Analysis:

- Validate assumptions.

- Evaluate the model using a confusion matrix and ROC curve.

- Interpret the results.

2. Perform a Decision Tree Analysis:

- Evaluate the model using a confusion matrix and ROC curve.

- Compare the decision tree results with the logistic regression results.

Assumptions Validation

Logistic regression assumes a linear relationship between the independent variables and the log odds of the dependent variable. It also requires the absence of multicollinearity and an appropriate.

Interpretation of Results

```
> confusion <- ConfusionMatrix(factor(pred_class), factor(test$Loan_Status))
> print(confusion)
      y_pred
y_true   0   1
     0  19  39
     1   4 123
```

- 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.

Loan Eligibility Prediction Analysis

Part B - Decision Tree Analysis

Interpretation of Results

```
      y_pred
y_true   0   1
     0  22  36
     1  15 112
```

- 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

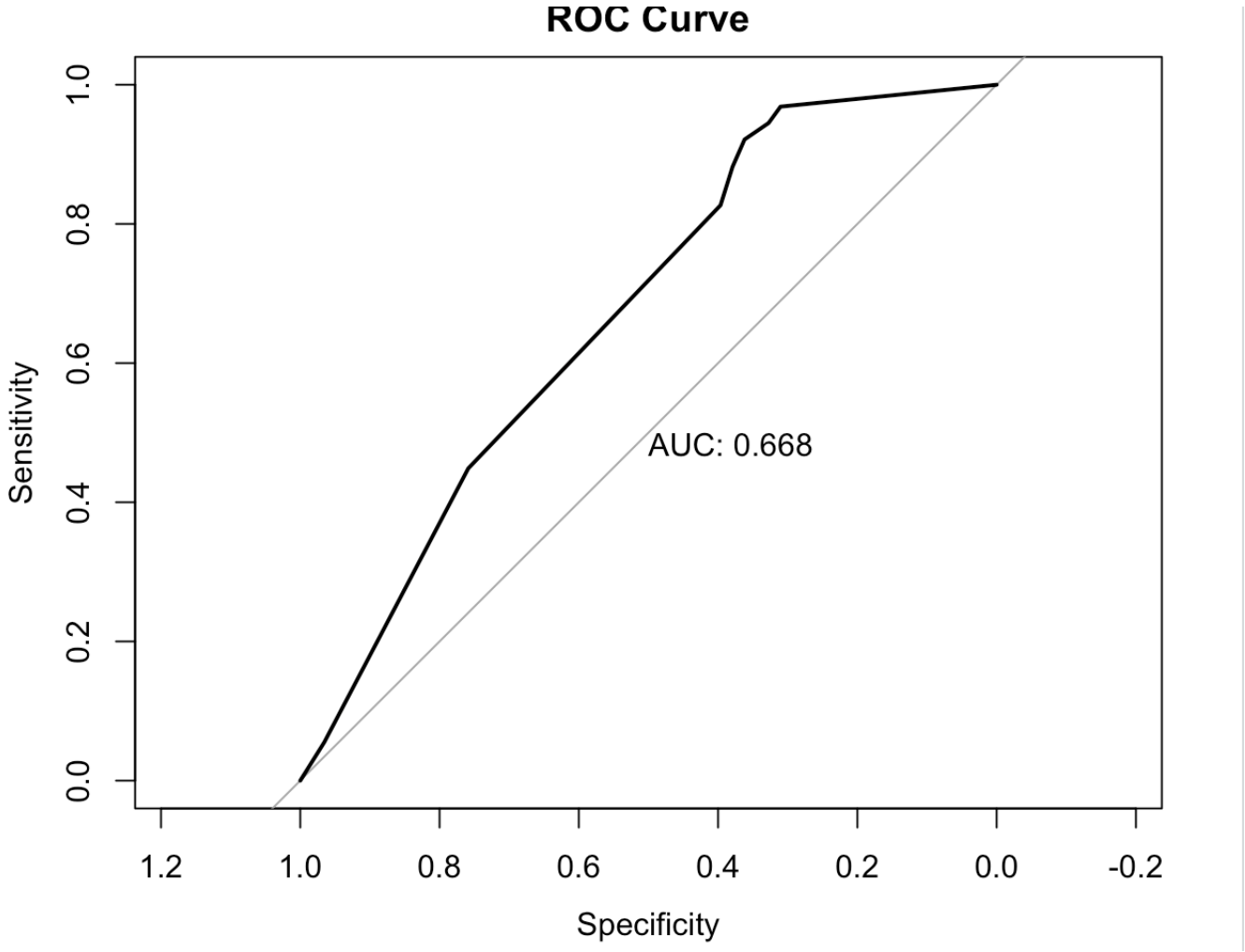
Conclusion

- 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.

Recommendations

1. **Feature Engineering:** Enhance input features to improve model performance.
2. **Model Tuning:** Optimize hyperparameters for better results.
3. **Cross-Validation:** Ensure consistent performance across different data splits.
4. **Explore Other Algorithms:** Try other models like Random Forest, Gradient Boosting, or SVM.

By addressing these recommendations, the predictive models' performance can be further improved, leading to better loan eligibility predictions.



Insights from the ROC Curve

The ROC (Receiver Operating Characteristic) curve is a graphical representation of a classifier's performance, plotting the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) at various threshold settings. The AUC (Area Under the Curve) quantifies the overall ability of the model to discriminate between positive and negative classes.

Key Points from the ROC Curve:

1. **AUC Value:**
   - The AUC value of 0.668 indicates that the decision tree model has a moderate ability to distinguish between eligible and non-eligible loan applicants.
   - An AUC of 0.5 suggests no discrimination (equivalent to random guessing), while an AUC of 1.0 indicates perfect discrimination.
   - Therefore, an AUC of 0.668 is better than random guessing but suggests that there is room for improvement in the model's discriminatory power.
2. **Curve Shape:**
   - The ROC curve starts at the origin (0,0) and moves towards the top left corner (1,1), where it achieves perfect classification.
   - The curve's initial steep ascent indicates a good trade-off between True Positive Rate (Sensitivity) and False Positive Rate.
   - However, the curve flattens out as it approaches the top right, indicating diminishing returns in sensitivity relative to an increase in false positives.
3. **Threshold Analysis:**
   - At low thresholds (rightmost part of the curve), the model classifies more instances as positive, increasing the True Positive Rate but also the False Positive Rate.
   - At high thresholds (leftmost part of the curve), the model becomes conservative, reducing both True Positives and False Positives.
   - The ideal balance depends on the specific cost of false positives versus false negatives in your application context.

Recommendations for Improvement:

1. **Feature Engineering:**
   - Enhance the input features by creating new features or transforming existing ones to improve predictive power.
   - Explore interactions between variables and non-linear relationships.
2. **Model Tuning:**
   - Optimize hyperparameters of the decision tree model, such as maximum depth, minimum samples per leaf, and criterion (e.g., Gini impurity vs. entropy).
   - Consider pruning the tree to prevent overfitting.
3. **Alternative Models:**
   - Try other classification algorithms like Random Forest, Gradient Boosting, or Support Vector Machines (SVM) that might capture more complex patterns in the data.
   - Ensemble methods can be particularly effective in improving model performance.
4. **Cross-Validation:**
   - Use cross-validation to ensure the model's performance is consistent across different subsets of the data.
   - This helps in getting a more reliable estimate of the model's performance and avoiding overfitting.
5. **Balancing the Dataset:**
   - If the dataset is imbalanced (i.e., the number of eligible vs. non-eligible applicants is skewed), use techniques like SMOTE (Synthetic Minority Over-sampling Technique) or undersampling the majority class.
   - This can help the model to learn better decision boundaries.
6. **Adjusting Decision Threshold:**
   - Experiment with different probability thresholds for classifying an applicant as eligible to find a better balance between precision and recall.

- This can be particularly useful if the costs of false positives and false negatives are not equal.

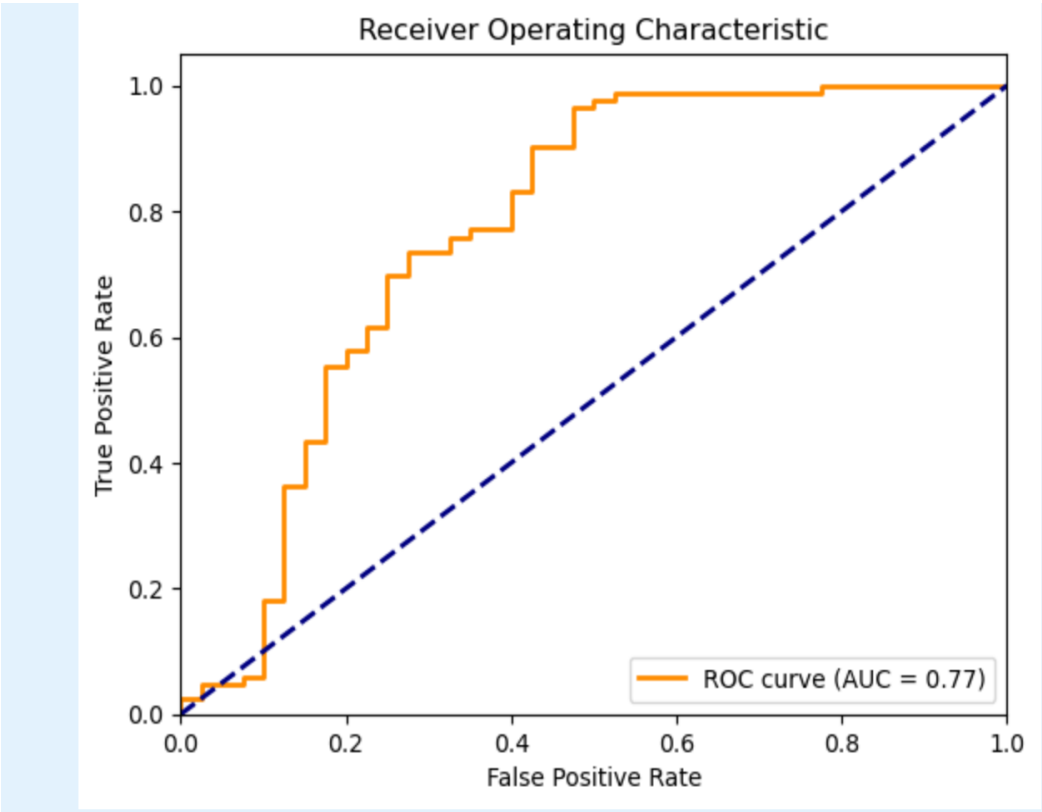
By implementing these strategies, you can potentially improve the model's performance and achieve a higher AUC-ROC value, indicating better discrimination between eligible and non-eligible loan applicants.

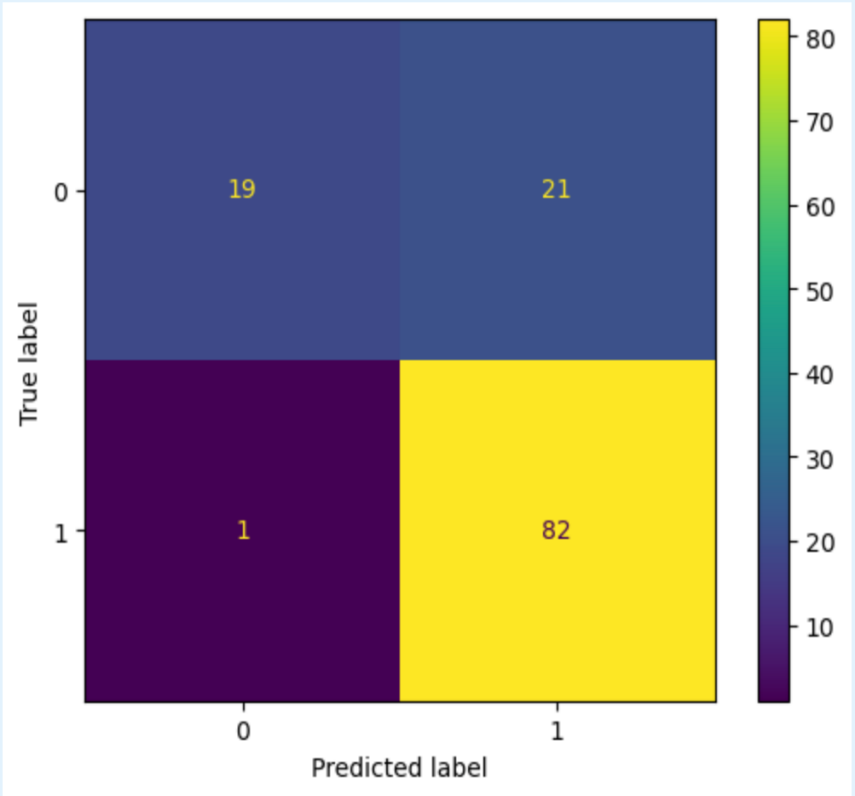
CODES

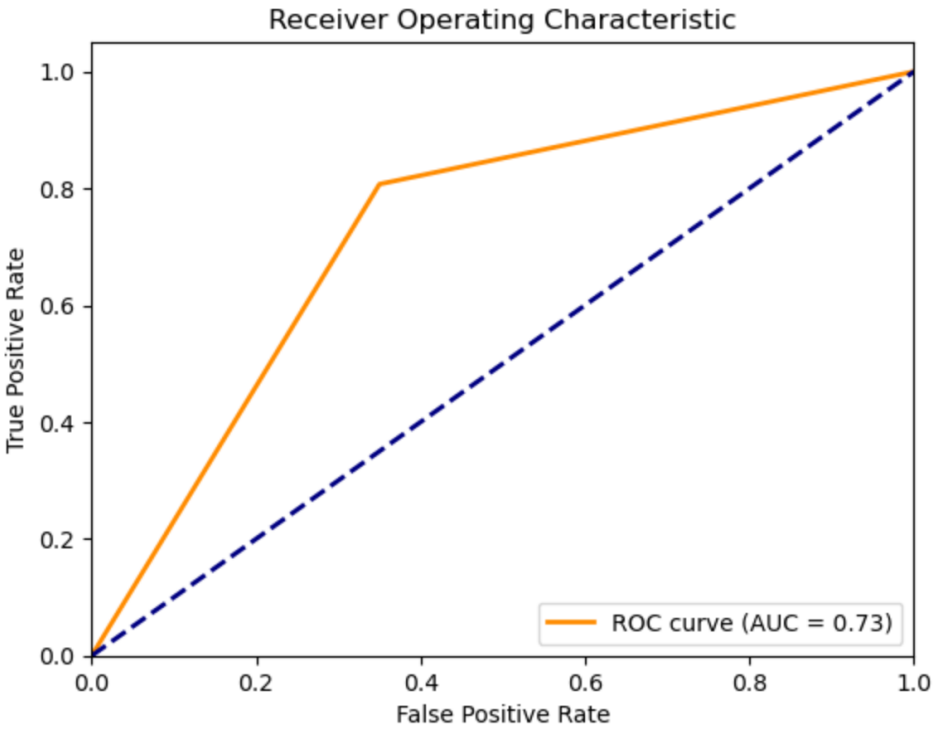
```
# Install necessary packages if not already installed
if (!require(caTools)) install.packages('caTools', dependencies=TRUE)
if (!require(MLmetrics)) install.packages('MLmetrics', dependencies=TRUE)
if (!require(pROC)) install.packages('pROC', dependencies=TRUE)
if (!require(rpart)) install.packages('rpart', dependencies=TRUE)
if (!require(rpart.plot)) install.packages('rpart.plot', dependencies=TRUE)

# Load necessary libraries
library(dplyr)
library(ggplot2)
library(DataExplorer)
library(caTools)
library(MLmetrics)
library(pROC)
library(rpart)
library(rpart.plot)

# Read the dataset
eligibility <- read.csv("/Users/janybalashiva/Downloads/Loan Eligibility Prediction.csv")

# Display the column names and structure of the dataset
names(eligibility)
str(eligibility)

# Check unique values in the Loan_Status column
unique(eligibility$Loan_Status)

# Assuming Loan_Status is not binary, convert it to binary (e.g., 'Y' -> 1, 'N' -> 0)
eligibility <- eligibility %>%
  mutate(Loan_Status = ifelse(Loan_Status == 'Y', 1, 0))

# Replace missing values with the mean for numeric columns
eligibility <- eligibility %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))

# Check for remaining missing values
missing_info <- colSums(is.na(eligibility))
cat("Missing Values Information:\n")
print(missing_info)

# Select only numeric columns for correlation matrix
numeric_columns <- eligibility %>%
  select(where(is.numeric))
```

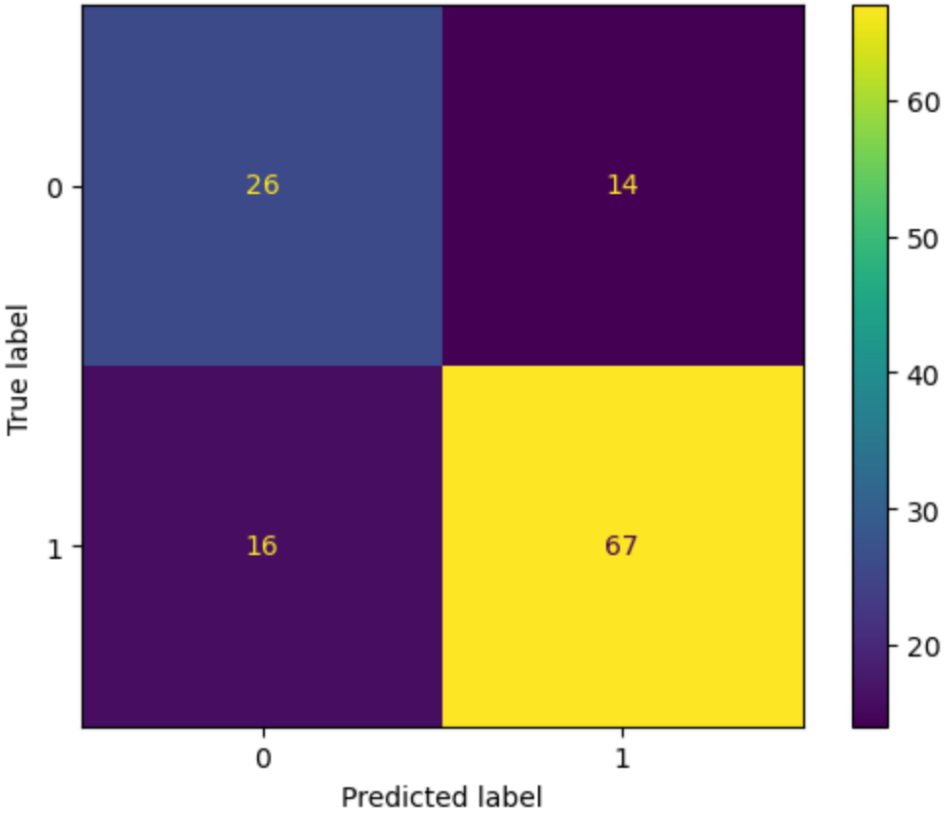
```
# Calculate and print correlation matrix
cor_matrix <- cor(numeric_columns)
print(cor_matrix)
heatmap(cor_matrix)

# Boxplot for numeric columns against Loan_Status
# Replace the following line with actual column names from your dataset
boxplot(Applicant_Income + Coapplicant_Income + Loan_Amount + Loan_Amount_Term +
Credit_History ~ Loan_Status, data=eligibility)

# Logistic Regression
set.seed(123)
split <- sample.split(eligibility$Loan_Status, SplitRatio = 0.7)
train <- subset(eligibility, split == TRUE)
test <- subset(eligibility, split == FALSE)
model <- glm(Loan_Status ~ ., data = train, family = binomial)
pred_prob <- predict(model, newdata=test, type="response")
pred_class <- ifelse(pred_prob >= 0.5, 1, 0)

# Confusion Matrix
confusion <- ConfusionMatrix(factor(pred_class), factor(test$Loan_Status))
print(confusion)
roc_obj <- roc(test$Loan_Status, pred_prob)
auc <- auc(roc_obj)
print(paste("AUC-ROC:", auc))
plot(roc_obj, main="ROC Curve", print.auc=TRUE)

# Decision tree analysis
set.seed(123)
split <- sample.split(eligibility$Loan_Status, SplitRatio = 0.7)
train <- subset(eligibility, split == TRUE)
test <- subset(eligibility, split == FALSE)
model <- rpart(Loan_Status ~ ., data=train, method="class")
pred_prob <- predict(model, newdata=test, type="prob")
pred_class <- ifelse(pred_prob[,2] >= 0.5, 1, 0)

# Confusion Matrix for decision tree
confusion <- ConfusionMatrix(factor(pred_class), factor(test$Loan_Status))
print(confusion)
roc_obj <- roc(test$Loan_Status, pred_prob[,2])
auc <- auc(roc_obj)
print(paste("AUC-ROC:", auc))
plot(roc_obj, main="ROC Curve", print.auc=TRUE)
```

PYTHON


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
ROC curve (AUC = 0.77)
0.0
0.2
0.4
0.6
0.8
1.0


19
21
1
82
True label
Predicted label
0
1
80
70
60
50
40
30
20
10


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (AUC = 0.73)


26
14
16
67
True label
Predicted label
0
1
60
50
40
30
20

Comparison Table

model	class	precision	recall	f1-score	support
Decision Tree	0	0.65	0.68	0.67	105
Decision Tree	1	0.80	0.78	0.79	175
Logistic Regression	0	0.81	0.47	0.59	105
Logistic Regression	1	0.82	0.96	0.88	175

Insights

Logistic Regression:

- **Precision and Recall:**
  - For class 1 (loan approved), the 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 0 (loan not approved), 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).
- **F1-Score:**
  - The F1-score for class 1 is 0.88, indicating strong performance in predicting loan approvals.
  - The F1-score for class 0 is lower at 0.59, indicating moderate performance in predicting loan rejections.
- **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.
- **F1-Score:**
  - The F1-score for class 1 is 0.79, indicating good performance in predicting loan approvals.
  - The F1-score for class 0 is 0.67, indicating moderate performance in predicting loan rejections.
- **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.

CODES

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.

```
Optimization terminated successfully.
         Current function value: 0.629775
         Iterations 4
                          Probit Regression Results
==============================================================================
Dep. Variable:                 non_veg   No. Observations:               101655
Model:                          Probit   Df Residuals:                   101651
Method:                            MLE   Df Model:                            3
Date:                Mon, 01 Jul 2024   Pseudo R-squ.:                0.001666
Time:                        22:55:49   Log-Likelihood:                -64020.
converged:                       True   LL-Null:                       -64127.
Covariance Type:            nonrobust   LLR p-value:                 4.613e-46
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.5686      0.017     32.573      0.000       0.534       0.603
Age           -0.0002      0.000     -0.749      0.454      -0.001       0.000
MPCE_URP   -2.932e-06   8.99e-07     -3.259      0.001   -4.69e-06   -1.17e-06
Education     -0.0154      0.001    -13.467      0.000      -0.018      -0.013
==============================================================================
```

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==============================================================================\nD
ep. 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.

```
Optimization terminated successfully.
         Current function value: 0.718814
         Iterations: 212
         Function evaluations: 352
                            Tobit Results
==============================================================================
Dep. Variable:                non_veg   Log-Likelihood:                -73071.
Model:                          Tobit   AIC:                         1.462e+05
Method:            Maximum Likelihood   BIC:                         1.462e+05
Date:                Mon, 01 Jul 2024
Time:                        19:01:26
No. Observations:              101655
Df Residuals:                  101651
Df Model:                           3
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0052      0.008      0.692      0.489      -0.010       0.020
Age            0.0107      0.000     82.978      0.000       0.010       0.011
MPCE_URP   -1.156e-06   3.76e-07     -3.075      0.002   -1.89e-06   -4.19e-07
Education      0.0210      0.000     46.020      0.000       0.020       0.022
par0           0.4964      0.001    397.637      0.000       0.494       0.499
==============================================================================
```

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_81973/2
23527810.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/ipykern
el_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":"Optim
ization terminated successfully.\n         Current function value: 0.718814\n
Iterations: 212\n         Function evaluations: 352\n
Tobit Results
\n==============================================================================\nD
ep. 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/anacond
a3/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":[]}]
```