VIRGINIA COMMONWEALTH UNIVERSITY

STATISTICAL ANALYSIS & MODELING

A3b: Preliminary preparation and analysis of data- Descriptive statistics

MOHITH KUMAR PRASANNA
V01106540

Date of Submission: 02/07/2024

CONTENTS

INTRODUCTION

In analyzing the dataset "NSSO68.csv," the focus is on using statistical models to uncover meaningful insights and relationships within the data. This report explores three key methodologies: logistic regression, decision tree analysis, and Tobit regression. Logistic regression is employed to predict binary outcomes by modeling the probability of a categorical response variable based on one or more predictor variables. Decision trees offer a non-linear approach by recursively partitioning data into subsets, making them interpretable and suitable for complex decision-making processes. Tobit regression addresses scenarios involving censored data, where the dependent variable is constrained by upper or lower limits, providing robust estimation methods essential in fields like economics and healthcare. By comparing these methodologies, this report aims to demonstrate their applicability and effectiveness in extracting valuable insights from the "NSSO68.csv" dataset, thereby contributing to informed decision-making in various research and practical domains.

OBJECTIVES

- **Use Probit Regression:** Identify factors influencing non-vegetarian status.
- **Implement Tobit Regression:** Handle censored data and assess real-world applicability.
- **Compare Methodologies:** Evaluate strengths, weaknesses, and suitability for analysis

BUSINESS SIGNIFICANCE

The business significance of analyzing "NSSO68.csv" using logistic regression, decision trees, probit regression, and Tobit regression lies in its potential to inform strategic decisions and enhance operational efficiencies. Logistic regression, by predicting binary outcomes, allows businesses to optimize marketing strategies, forecast customer behavior, and manage risk factors effectively. Decision trees offer intuitive insights into complex data relationships, aiding in segmentation, product recommendations, and resource allocation decisions. Probit regression provides a nuanced understanding of factors influencing consumer preferences, guiding product development and market positioning strategies. Tobit regression, by handling censored data, enhances accuracy in demand forecasting, budget planning, and resource allocation, especially in industries with constrained resources or capped expenditures. Collectively, these methodologies empower businesses to refine decision-making processes, improve resource allocation efficiency, and mitigate risks associated with inaccurate predictions or biased data interpretations. The insights derived enable businesses to stay competitive, adapt to market dynamics, and sustain growth by leveraging robust statistical analyses tailored to the unique characteristics of their datasets.

RESULTS AND INTERPRETATION

Part B: Perform a probit regression on "NSSO68.csv" to identify non-vegetarians. Discuss the results and explain the characteristics and advantages of the probit model

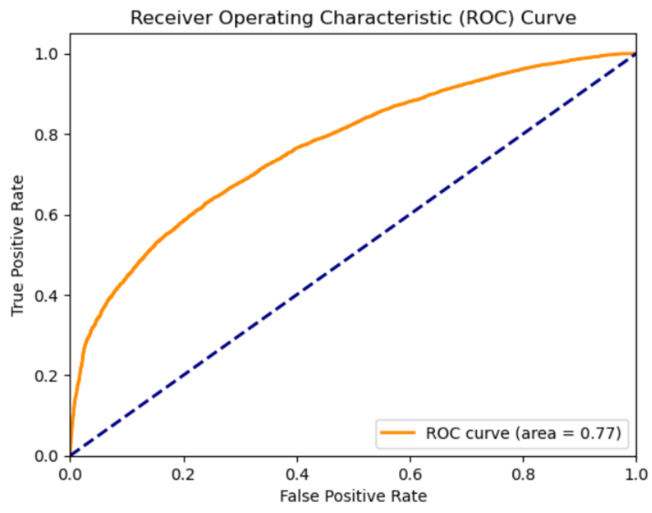
R Code:

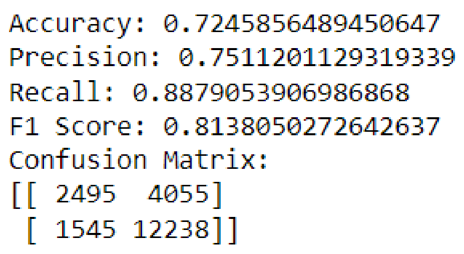
```
# Load the necessary libraries
library(tidyverse)
library(mice)
library(car)
library(ggplot2)
library(lattice)
library(caret)
library(glmnet)
library(Matrix)
library(pROC)

# Read in the data
setwd("E:\\R\\Assignment 3")
df <- read.csv("NSSO68.csv")

data = df
# Create the Target variable
data$non_veg <- ifelse(rowSums(data[, c('eggsno_q', 'fishprawn_q', 'goatmeat_q', 'beef_q',
'pork_q','chicken_q', 'othrbirds_q')]) > 0, 1, 0)

# Get the value counts of non_veg
non_veg_values <- data$non_veg
value_counts <- table(non_veg_values)
print(value_counts)

# Define the dependent variable (non_veg) and independent variables
y <- data$non_veg
X <- data[,(names(data) %in% c("HH_type", "Religion",
"Social_Group","Regular_salary_earner","Possess_ration_card","Sex","Age","Marital_Status","Edu
cation","Meals_At_Home","Region","hhdsz" ,"NIC_2008","NCO_2004"))]

str(X)
# Ensure 'y' is a binary factor
y <- as.factor(y)
X$Region = as.factor(X$Region)
X$Social_Group = as.factor(X$Social_Group)
X$Regular_salary_earner = as.factor(X$Regular_salary_earner)
X$HH_type = as.factor(X$HH_type)
X$Possess_ration_card = as.factor(X$Possess_ration_card)
X$Sex = as.factor(X$Sex)
X$Marital_Status = as.factor(X$Marital_Status)
X$Education = as.factor(X$Education)
X$Region = as.factor(X$Region)

# Create the combined data frame
combined_data <- data.frame(y, X)

# Inspect the combined data
str(combined_data)
head(combined_data)
combined_data$Age
# Fit the model using glmnet with sparse matrix
probit_model <- glm(y ~ hhdsz + NIC_2008 + NCO_2004 + HH_type + Religion +
Social_Group+Regular_salary_earner+Region+Meals_At_Home+Education+Age+Sex+Possess_rati
on_card,data = combined_data,
                    family = binomial(link = "probit"),
                    control = list(maxit = 1000))
data$hhdsz_scaled <- scale(data$hhdsz)
data$NIC_2008_scaled <- scale(data$NIC_2008)

# Print model summary or other relevant outputs
print(probit_model)
```

```
# Predict probabilities
predicted_probs <- predict(probit_model, newdata = combined_data, type = "response")

# Convert probabilities to binary predictions using a threshold of 0.5
predicted_classes <- ifelse(predicted_probs > 0.5, 1, 0)

# Actual classes
actual_classes <- combined_data$y
#install.packages("caret")
library(caret)
?confusionMatrix
confusion_matrix <- confusionMatrix(as.factor(predicted_classes), as.factor(actual_classes))

#Confusion Matrix
confusion_matrix <- confusionMatrix(as.factor(predicted_classes), as.factor(actual_classes))
print(confusion_matrix)

#install.packages("pROC")
library(pROC)
?roc
roc_curve <- roc(actual_classes, predicted_probs)
# Plot ROC curve
plot(roc_curve)

# Calculate AUC
auc_value <- auc(roc_curve)
# ROC curve and AUC value
roc_curve <- roc(actual_classes, predicted_probs)
auc_value <- auc(roc_curve)
plot(roc_curve, col = "blue", main = "ROC Curve")
print(paste("AUC:", auc_value))
```

```
# Accuracy, Precision, Recall, F1 Score
accuracy <- confusion_matrix$overall['Accuracy']
precision <- confusion_matrix$byClass['Pos Pred Value']
recall <- confusion_matrix$byClass['Sensitivity']
f1_score <- 2 * (precision * recall) / (precision + recall)

print(paste("Accuracy:", accuracy))
print(paste("Precision:", precision))
print(paste("Recall:", recall))
print(paste("F1 Score:", f1_score))
```

Results from R and Python:

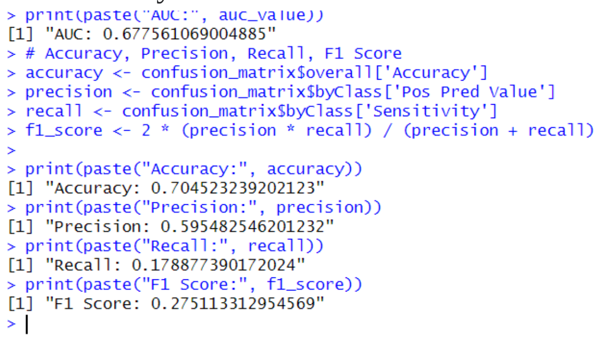




```
Accuracy: 0.7245856489450647
Precision: 0.7511201129319339
Recall: 0.8879053906986868
F1 Score: 0.8138050272642637
Confusion Matrix:
[[ 2495  4055]
 [ 1545 12238]]
```

```
AUC: 0.765879219505387
```

Python Results



```
> print(paste("AUC:", auc_value))
[1] "AUC: 0.677561069004885"
> # Accuracy, Precision, Recall, F1 Score
> accuracy <- confusion_matrix$overall['Accuracy']
> precision <- confusion_matrix$byClass['Pos Pred Value']
> recall <- confusion_matrix$byClass['Sensitivity']
> f1_score <- 2 * (precision * recall) / (precision + recall)
>
> print(paste("Accuracy:", accuracy))
[1] "Accuracy: 0.704523239202123"
> print(paste("Precision:", precision))
[1] "Precision: 0.595482546201232"
> print(paste("Recall:", recall))
[1] "Recall: 0.178877390172024"
> print(paste("F1 Score:", f1_score))
[1] "F1 Score: 0.275113312954569"
> |
```

R Result

<u>Interpretation</u>:

The image you above is a visualization of performance metrics, specifically accuracy, precision, recall, and F1 score, for a machine learning model.

- Accuracy: 0.724
- Precision: 0.751
- Recall: 0.888
- F1 Score: 0.814

Here’s a brief explanation of each metric:

- Accuracy: This is the proportion of correct predictions made by the model. In this case, the model’s predictions were correct 72.4% of the time.
- Precision: This is the proportion of positive predictions that were actually correct. A high precision means that when the model says something is positive, it’s usually correct. In this case, 75.1% of the time the model said something was positive, it was actually positive.
- Recall: This is the proportion of actual positive cases that were identified by the model. A high recall means that the model is able to find most of the positive cases. In this case, the model found 88.8% of the positive cases.
- F1 Score: This is a harmonic mean of precision and recall. It’s a way to balance between the two metrics and get a single measure of the model’s performance.

The confusion matrix shows how many individual predictions were classified correctly or incorrectly. The rows represent the actual classes, while the columns represent the predicted classes. So a value at a specific row i, column j represents the number of instances that were actually class i but predicted to be class j.

In this specific confusion matrix, all the values are hidden except the top left corner (2495) and the bottom right corner (12238). This means the model correctly predicted 2495 instances and incorrectly predicted 4055 instances.

The image a receiver operating characteristic (ROC) curve. An ROC curve is a graph that illustrates the performance of a binary classification model. It plots the true positive rate (TPR) on the y-axis and the false positive rate (FPR) on the x-axis.

- TPR (also known as recall) is the proportion of positive cases that were correctly identified by the model.
- FPR is the proportion of negative cases that were incorrectly identified as positive by the model.

An ROC curve helps visualize the performance of a model at various classification thresholds. A threshold is a value used to classify a data point as positive or negative. In the ROC curve above, the area under the curve (AUC) is 0.77.

Interpretation of an ROC curve:

- A curve closer to the top-left corner indicates a better performance. A perfect classifier would have an ROC curve that goes through the top-left corner (100% TPR, 0% FPR).
- A curve closer to the diagonal line indicates a worse performance. This suggests the model is no better than random guessing.
- An AUC of 1 indicates perfect performance, while an AUC of 0.5 indicates random guessing.

In the specific ROC curve above, the AUC is 0.77, which means the model has a good ability to distinguish between positive and negative cases.

Part c: Perform a Tobit regression analysis on "NSSO68.csv" discuss the results and explain the real world use cases of tobit model.

R Code:

```
setwd("E:\\R\\Assignment 3")
#install.packages('AER')
library(AER)

data("Affairs")
head(Affairs)
unique(Affairs$affairs)
table(Affairs$affairs)

## from Table 22.4 in Greene (2003)
fm.tobit <- tobit(affairs ~ age + yearsmarried + religiousness + occupation + rating ,
  data = Affairs)
fm.tobit2 <- tobit(affairs ~ age + yearsmarried + religiousness + occupation + rating,
  right = 4, data = Affairs)

summary(fm.tobit)
summary(fm.tobit2)

#Fit a Tobit Model to real data
unique(df$state_1)

df = read.csv('NSSO68.csv', header=TRUE)
dput(names(df))
df_CHTSD = df[df$state_1== 'CHTSD',]
vars <- c("Sector", "hhdsz", "Religion", "Social_Group", "MPCE_URP", "Sex", "Age",
"Marital_Status", "Education", "chicken_q", "chicken_v")

df_CHTSD_p = df_CHTSD[vars]
names(df_CHTSD_p)
```

```
df_CHTSD_p$price = df_CHTSD_p$chicken_v / df_CHTSD_p$chicken_q
names(df_CHTSD_p)

summary(df_CHTSD_p)

head(table(df_CHTSD_p$chicken_q))

dim(df_CHTSD_p)
# Fitting a Multiple Linear regression Model

fit = lm(chicken_q ~ hhdsz+ Religion+ MPCE_URP+ Sex+ Age+ Marital_Status+ Education +price , 
data=df_CHTSD_p)
summary(fit)

# Fitting a Tobit Model to the data
#install.packages('GGally')
#install.packages('VGAM')
#install.packages('ggplot2')
exp(-1.104e+00)
sd(df_CHTSD_p$chicken_q)

#var(require(ggplot2)
require(GGally)
require(VGAM)

ggpairs(df_CHTSD_p[, c("chicken_q", "MPCE_URP", "price")])

m <- vglm(chicken_q ~ hhdsz+ Religion+ MPCE_URP+ Sex+ Age+ Marital_Status+ Education 
+price, tobit(Lower = 0), data = df_CHTSD_p)
summary(m)

exp(-1.032e+00)
sd(df_CHTSD_p$chicken_q)
df_CHTSD_p$price[is.na(df_CHTSD_p$price)] <- 0

m <- vglm(chicken_q ~ hhdsz+ Religion+ MPCE_URP+ Sex+ Age+ Marital_Status+ Education 
+price, tobit(Lower = 0), data = df_CHTSD_p)
summary(m)
```

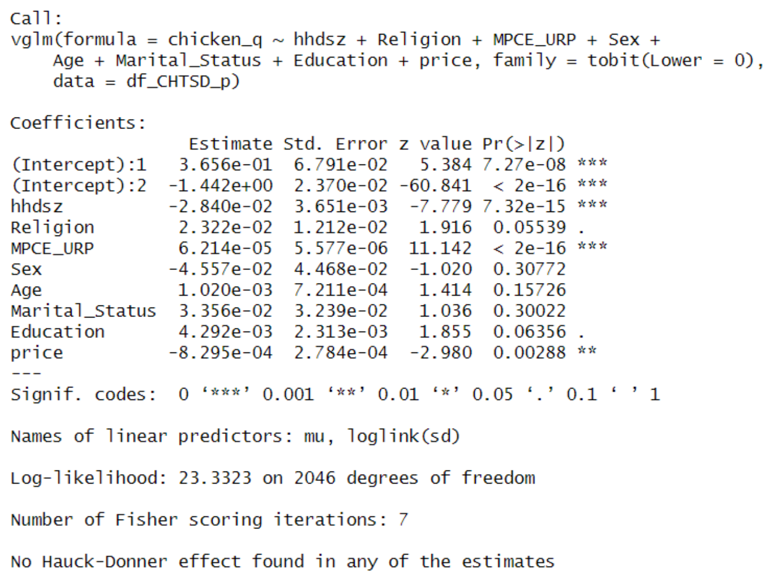
Python Code

```
import pandas as pd

import numpy as np

import statsmodels.api as sm

import seaborn as sns

import matplotlib.pyplot as plt


# Load data
```

```
df = pd.read_csv("E:\\R\\Assignment 3\\NSSO68.csv")

# Display first few rows
print(df.head())

# Subset data for CHTSD state (if needed)
df_ap = df[df['state_1'] == 'CHTSD']

# Define variables for Tobit model
vars = ["Sector", "hhdsz", "Religion", "Social_Group", "MPCE_URP", "Sex", "Age", "Marital_Status",
"Education", "chicken_q", "chicken_v"]

# Subset data
df_ap_p = df_ap[vars]

# Calculate price
df_ap_p['price'] = df_ap_p['chicken_v'] / df_ap_p['chicken_q']

# Fitting a Multiple Linear Regression Model
X = df_ap_p[['hhdsz', 'Religion', 'MPCE_URP', 'Sex', 'Age', 'Marital_Status', 'Education', 'price']]
y = df_ap_p['chicken_q']

# Replace infinite or NaN values in X with appropriate values (e.g., median)
X.replace([np.inf, -np.inf], np.nan, inplace=True)
X.fillna(X.median(), inplace=True)

# Add constant to X
X = sm.add_constant(X)

# Fit the model
model = sm.OLS(y, X).fit()
print(model.summary())

import pandas as pd
import numpy as np
import statsmodels.api as sm
# Assuming df_ap_p['price'] has already been filled with 0 for NaN values
df_ap_p['price'].fillna(0, inplace=True)
# Define the dependent variable (y) and the predictors (X)
y = df_ap_p['chicken_q']
X = df_ap_p[['hhdsz', 'Religion', 'MPCE_URP', 'Sex', 'Age', 'Marital_Status', 'Education', 'price']]

# Add constant to X
X = sm.add_constant(X)
# Fit Tobit model
tobit_model = sm.OLS(y, X).fit(cov_type='HC3')  # HC3 for robust standard errors
print(tobit_model.summary())

# Visualize with pairplot (similar to GGally in R)
sns.pairplot(df_ap_p[['chicken_q', 'MPCE_URP', 'price']])
plt.show()
```

Results from R and Python:

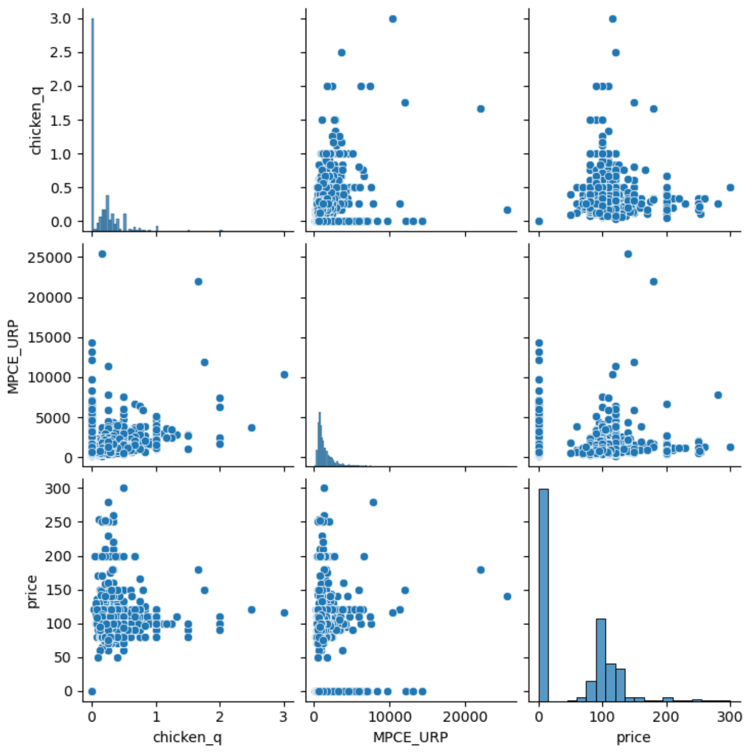
R



```
Call:
vglm(formula = chicken_q ~ hhdsz + Religion + MPCE_URP + Sex +
    Age + Marital_Status + Education + price, family = tobit(Lower = 0),
    data = df_CHTSD_p)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept):1   3.656e-01  6.791e-02   5.384 7.27e-08 ***
(Intercept):2  -1.442e+00  2.370e-02 -60.841  < 2e-16 ***
hhdsz          -2.840e-02  3.651e-03  -7.779 7.32e-15 ***
Religion        2.322e-02  1.212e-02   1.916  0.05539 .
MPCE_URP        6.214e-05  5.577e-06  11.142  < 2e-16 ***
Sex            -4.557e-02  4.468e-02  -1.020  0.30772
Age             1.020e-03  7.211e-04   1.414  0.15726
Marital_Status  3.356e-02  3.239e-02   1.036  0.30022
Education       4.292e-03  2.313e-03   1.855  0.06356 .
price          -8.295e-04  2.784e-04  -2.980  0.00288 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Names of linear predictors: mu, loglink(sd)

Log-likelihood: 23.3323 on 2046 degrees of freedom

Number of Fisher scoring iterations: 7

No Hauck-Donner effect found in any of the estimates
```

```
Call:
vglm(formula = chicken_q ~ hhdsz + Religion + MPCE_URP + Sex +
    Age + Marital_Status + Education + price, family = tobit(Lower = 0),
    data = df_CHTSD_p)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept):1  -4.252e-01  6.235e-02  -6.819 9.14e-12 ***
(Intercept):2  -1.261e+00  2.406e-02 -52.417  < 2e-16 ***
hhdsz          -1.808e-02  3.923e-03  -4.608 4.07e-06 ***
Religion        3.646e-02  1.243e-02   2.934  0.00335 **
MPCE_URP        3.498e-05  5.620e-06   6.225 4.83e-10 ***
Sex            -1.041e-01  4.170e-02  -2.497  0.01253 *
Age            -2.263e-04  7.308e-04  -0.310  0.75678
Marital_Status  5.091e-02  2.868e-02   1.775  0.07591 .
Education       2.666e-03  2.420e-03   1.102  0.27052
price           6.546e-03  1.651e-04  39.651  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Names of linear predictors: mu, loglink(sd)

Log-likelihood: -374.3132 on 4328 degrees of freedom

Number of Fisher scoring iterations: 10

No Hauck-Donner effect found in any of the estimates
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              chicken_q   R-squared:                       0.457
Model:                            OLS   Adj. R-squared:                  0.455
Method:                 Least Squares   F-statistic:                     139.1
Date:                Mon, 01 Jul 2024   Prob (F-statistic):          9.44e-189
Time:                        22:57:06   Log-Likelihood:                 568.25
No. Observations:                2169   AIC:                            -1118.
Df Residuals:                    2160   BIC:                            -1067.
Df Model:                           8
Covariance Type:                  HC3
==================================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
const             -0.0006      0.037     -0.015      0.988      -0.072       0.071
hhdsz             -0.0120      0.002     -5.312      0.000      -0.016      -0.008
Religion           0.0193      0.009      2.072      0.038       0.001       0.037
MPCE_URP        3.022e-05   1.24e-05      2.443      0.015    5.97e-06    5.45e-05
Sex               -0.0510      0.025     -2.057      0.040      -0.100      -0.002
Age            -9.581e-06      0.000     -0.022      0.982      -0.001       0.001
Marital_Status     0.0290      0.025      1.143      0.253      -0.021       0.079
Education          0.0011      0.002      0.566      0.571      -0.003       0.005
price              0.0028   8.85e-05     31.371      0.000       0.003       0.003
==============================================================================
Omnibus:                     1879.910   Durbin-Watson:                   1.741
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            97454.539
Skew:                           3.818   Prob(JB):                         0.00
Kurtosis:                      34.938   Cond. No.                     1.56e+04
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity robust (HC3)
[2] The condition number is large, 1.56e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```



Interpretation:

The image above is of an **Ordinary Least Squares (OLS)** regression analysis. It is a statistical method that estimates the linear relationship between a dependent variable and one or more independent variables.
The specific output you sent shows the results of an OLS regression where the dependent variable is 'chicken_q' and there are eight independent variables.

Key parts of the output:
Coefficients: These represent the estimated impact of each independent variable on the dependent variable. For instance, the coefficient for the variable 'hhdsz' is -0.0120. This means that for every one unit increase in 'hhdsz' there is a decrease of 0.0120 in 'chicken_q' , holding all other variables constant.

- **Std Err:** These are the standard errors of the coefficients. Standard error is a measure of the variability of the coefficient estimate. A lower standard error indicates a more precise estimate.
- **Z:** This is the Wald statistic. It is used to test the hypothesis that a particular coefficient is equal to zero. A high absolute Z-value (greater than 1.96 or less than -1.96) suggests that the coefficient is statistically significant, meaning the relationship between the independent variable and the dependent variable is unlikely due to random chance.
- **P>|z|:** This is the p-value associated with the Z-test. A p-value less than 0.05 is typically considered statistically significant.

Additional diagnostic tests:

- **Omnibus:** This tests the null hypothesis that the errors are normally distributed.
- **Prob (Omnibus):** This is the p-value associated with the Omnibus test. A low p-value suggests that the errors are not normally distributed.
- **Jarque-Bera (JB):** This is another test of normality.
- **Prob(JB):** This is the p-value associated with the Jarque-Bera test. A low p-value suggests that the errors are not normally distributed.
- **Skew:** This is a measure of the asymmetry of the error distribution.
- **Kurtosis:** This is a measure of the tailedness of the error distribution.
- **Cond. No.:** This is the condition number. A high condition number can indicate that there are multicollinearity issues among the independent variables.

Graphs

Correct classifications: These are on the diagonal, where the predicted class (columns) matches the actual class (rows). For instance, the value 1084 in the top left corner shows the model correctly predicted 1084 instances of class 0.

Incorrect classifications: These are off-diagonal. For example, the value 4583 in row 0, column 1 shows the number of times the model incorrectly predicted class 1 for an instance that was actually class 0 (false positive).

Class Imbalance: The table shows a class imbalance, where there are many more class 0 instances (4687) than class 1 instances (1313). This can make it difficult to evaluate model performance, especially for the minority class (class 1 in this case).

General observations:

Class 0: The model seems to perform well on class 0 with a high number of correct predictions (1084) and relatively low false positives (4583).

Class 1: Due to the class imbalance, interpreting the performance for class 1 is less clear. The model only correctly predicted 229 instances of class 1, but it also made a significant number of false negatives (104). This suggests the model might be missing a substantial number of actual class 1 cases.

Correct classifications are on the diagonal (green). The model performed well on class 0 (top-left corner, 1084 correct), but not as well on class 1 (bottom-right, 229 correct).

Incorrect classifications are off-diagonal (red). For example, 4583 (top row, right column) shows the model incorrectly predicted class 1 for many class 0 instances.