Fall 2023 DATA 240 Data Mining

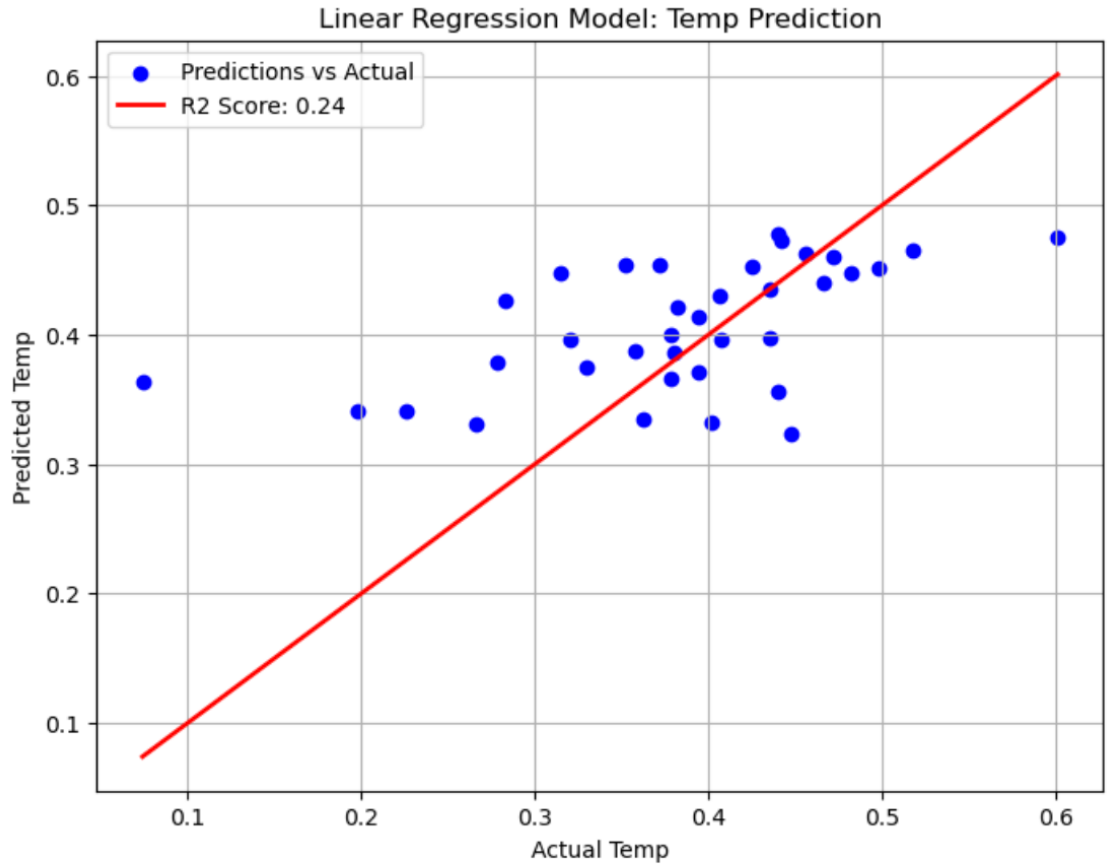
Homework – 2

Name :- Prayag Nikul Purani

SJSU Id :- 017416737

Question 1:

1a.




```
R-squared score: 0.23672098607947067
```

The scatter plot shows a **moderate spread** of the blue dots around the red line. While some points are close to the line, indicating decent predictions, many points are further away, especially for lower actual temperatures. This explains the low R-squared score. The model is **not very good** at capturing the relationship between the input features and the temperature, as shown by both the scatter of points and the R-squared score of **0.24**.

1b.

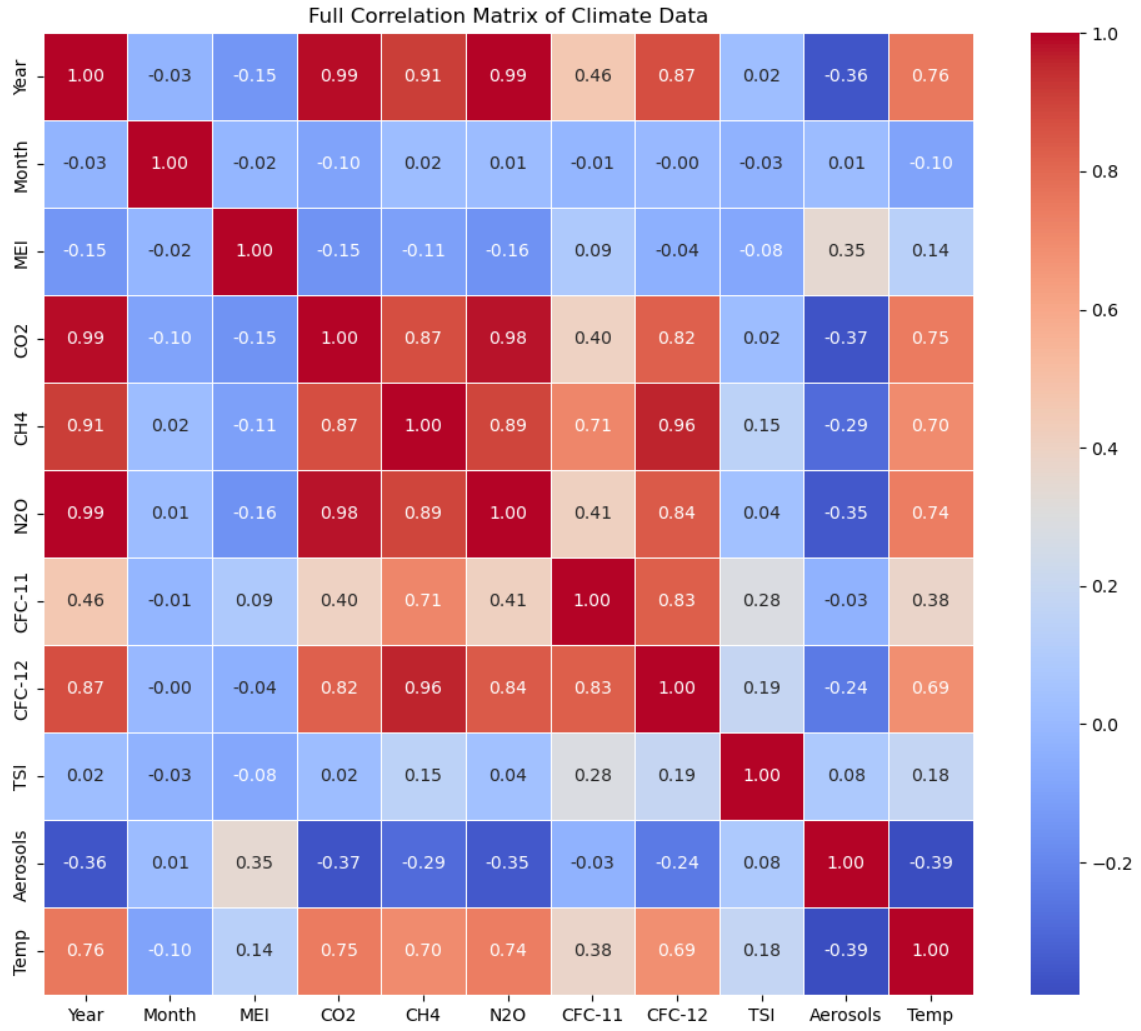
To represent the linear regression model as an equation, we can write it in the following form:

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Where:

- $\hat{y}$ is the predicted temperature (`Temp`).
- β_0 is the intercept (the constant term).
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for each independent variable.
- $X_1, X_2, \ldots, X_n$ represent the independent variables (e.g., `MEI`, `CO2`, `CH4`, etc.).

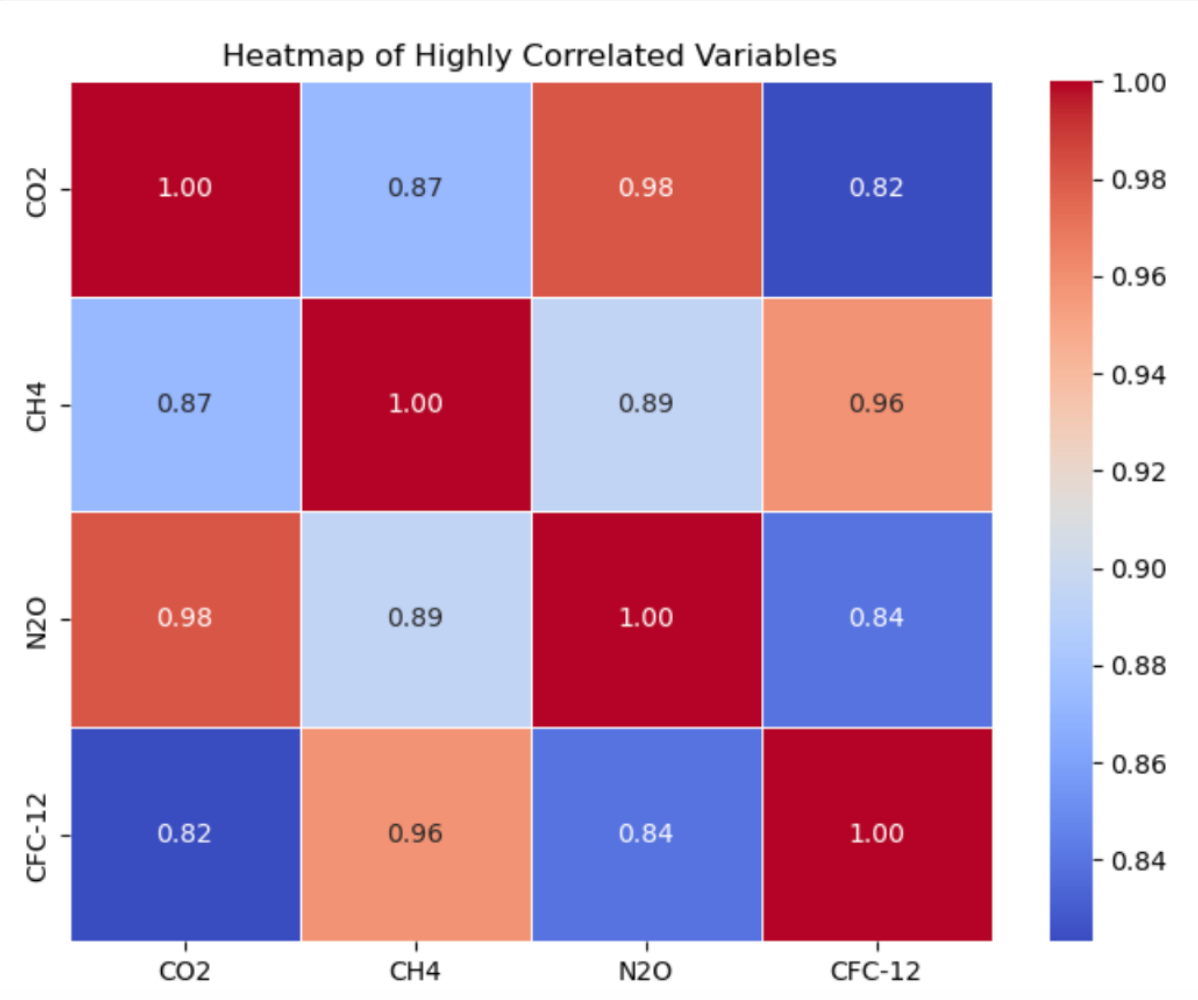
1.c



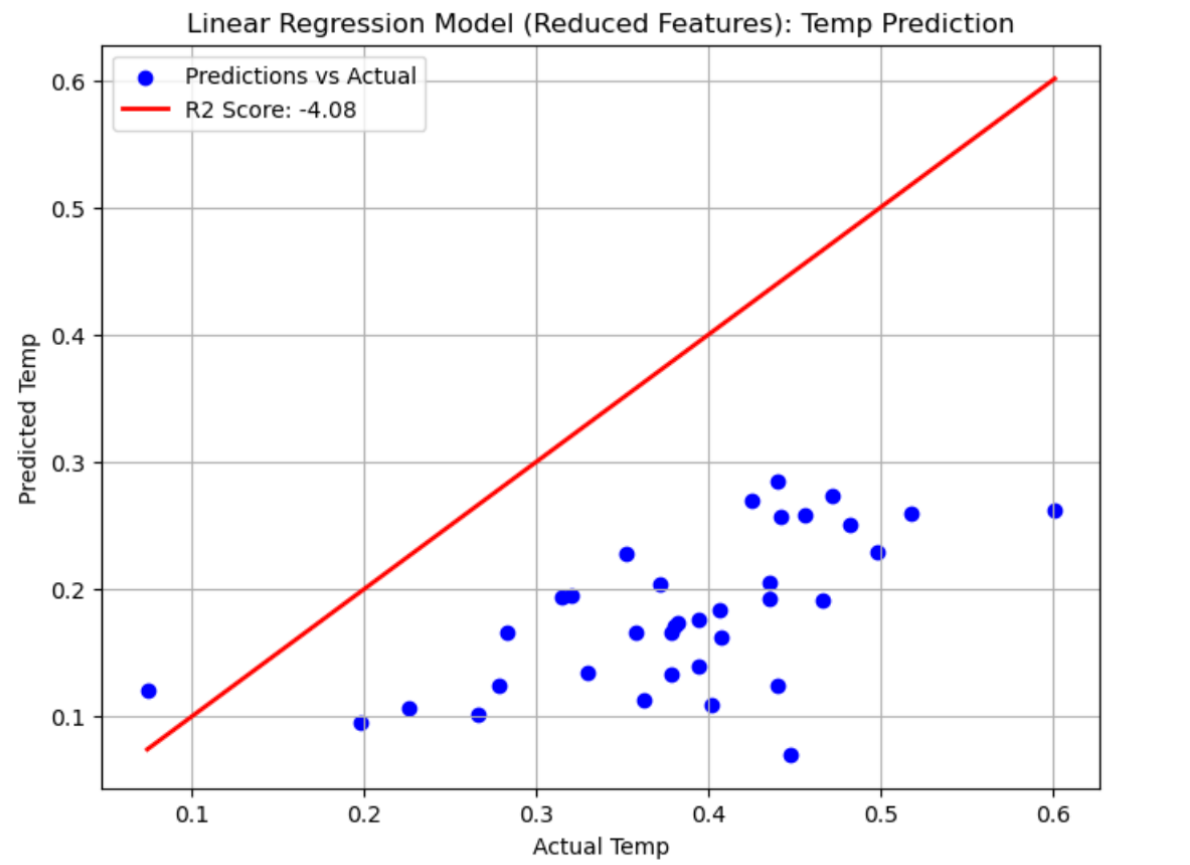
```
high_correlation_threshold = 0.75
highly_correlated_vars = correlation_matrix_full[(correlation_matrix_full > high_correlation_threshold) | (correlation_matrix_full < -hi
print("Highly Correlated Variables (correlation > 0.75 or < -0.75):")
print(highly_correlated_vars)
```

```
Highly Correlated Variables (correlation > 0.75 or < -0.75):
              Year  Month  MEI       CO2       CH4       N2O    CFC-11  \
Year      1.000000    NaN  NaN  0.985379  0.910563  0.994850       NaN
Month          NaN    1.0  NaN       NaN       NaN       NaN       NaN
MEI            NaN    NaN  1.0       NaN       NaN       NaN       NaN
CO2       0.985379    NaN  NaN  1.000000  0.872253  0.981135       NaN
CH4       0.910563    NaN  NaN  0.872253  1.000000  0.894409       NaN
N2O       0.994850    NaN  NaN  0.981135  0.894409  1.000000       NaN
CFC-11         NaN    NaN  NaN       NaN       NaN       NaN  1.000000
CFC-12    0.870067    NaN  NaN  0.823210  0.958237  0.839295  0.831381
TSI            NaN    NaN  NaN       NaN       NaN       NaN       NaN
Aerosols       NaN    NaN  NaN       NaN       NaN       NaN       NaN
Temp      0.755731    NaN  NaN       NaN       NaN       NaN       NaN

             CFC-12  TSI  Aerosols      Temp
Year      0.870067  NaN       NaN  0.755731
Month          NaN  NaN       NaN       NaN
MEI            NaN  NaN       NaN       NaN
CO2       0.823210  NaN       NaN       NaN
CH4       0.958237  NaN       NaN       NaN
N2O       0.839295  NaN       NaN       NaN
CFC-11    0.831381  NaN       NaN       NaN
CFC-12    1.000000  NaN       NaN       NaN
TSI            NaN  1.0       NaN       NaN
Aerosols       NaN  NaN       1.0       NaN
Temp           NaN  NaN       NaN  1.000000
```



1d.




```
R-squared score: -4.081440694045465
```

Comparison to Previous Plot:

- The previous model, with an R^2 of **0.24**, was far from perfect, but it still captured some variance in the temperature data.
- In contrast, the **reduced feature model** here performs much worse, as shown by the R-squared value of **-4.08**, indicating that removing certain features significantly harmed the model's predictive power.

Interpretation:

- **Overfitting or Loss of Predictive Power**: Removing correlated features may have led to the loss of important predictors, causing the model to perform poorly on unseen data.
- **Highly correlated features** might have been crucial for the original model's performance, and dropping them may have resulted in a lack of essential information for making good predictions.

Possible Actions to Improve:

- **Feature selection**: Instead of simply dropping correlated features, other methods like **regularization (Lasso, Ridge)** or **PCA** could help reduce feature multicollinearity while retaining important information.
- **Model complexity**: Explore other models that may better capture the relationships in the data, such as decision trees, random forests, or gradient boosting.

This plot and the resulting R-squared score indicate that this particular feature reduction has led to a **deterioration in model performance**.

Question 2:

➔ Load Dataset

```
In [2]: # Load the dataset
        data = pd.read_csv('asthma_sample_data.csv')
```

```
In [3]: # Display the first few rows to understand the data structure
        print("First few rows of the dataset:")
        print(data.head())
```

```
First few rows of the dataset:
   gender res_inf  ghq12  attack
0  female     yes     21       6
1    male      no     17       4
2    male     yes     30       8
3  female     yes     22       5
4    male     yes     27       2
```

➔ Convert categorical values to numerical values

```
In [4]: # Convert categorical variables to numerical format for modeling
        data['gender_num'] = np.where(data['gender'] == 'male', 1, 0)
        data['res_inf_num'] = np.where(data['res_inf'] == 'yes', 1, 0)
```

```
In [5]: # Define the dependent variable (count of asthma attacks)
        y = data['attack']
```

➔ Dependent variable

```
In [6]: # Model 1: Poisson regression using 'gender' as predictor
        X_gender = sm.add_constant(data['gender_num'])
        model_gender = sm.GLM(y, X_gender, family=sm.families.Poisson()).fit()
        print("\nPoisson Regression Model using 'gender' as predictor:")
        print(model_gender.summary())
```

```
Poisson Regression Model using 'gender' as predictor:
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                 attack   No. Observations:                  120
Model:                            GLM   Df Residuals:                      118
Model Family:                 Poisson   Df Model:                            1
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -248.15
Date:                Tue, 01 Oct 2024   Deviance:                       223.23
Time:                        21:35:28   Pearson chi2:                     191.
No. Iterations:                     4   Pseudo R-squ. (CS):           0.05136
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.0211      0.073     13.925      0.000       0.877       1.165
gender_num    -0.3000      0.121     -2.487      0.013      -0.536      -0.064
==============================================================================
```

2a

➔ Model 1 gender as predictor

In [7]:
```
# Model 2: Poisson regression using 'res_inf' as predictor
X_res_inf = sm.add_constant(data['res_inf_num'])
model_res_inf = sm.GLM(y, X_res_inf, family=sm.families.Poisson()).fit()
print("\nPoisson Regression Model using 'res_inf' as predictor:")
print(model_res_inf.summary())
```

```
Poisson Regression Model using 'res_inf' as predictor:
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                 attack   No. Observations:                  120
Model:                            GLM   Df Residuals:                      118
Model Family:                 Poisson   Df Model:                            1
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -226.78
Date:                Tue, 01 Oct 2024   Deviance:                       180.49
Time:                        21:35:28   Pearson chi2:                     168.
No. Iterations:                     5   Pseudo R-squ. (CS):             0.3356
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2877      0.121      2.372      0.018       0.050       0.525
res_inf_num    0.9032      0.138      6.533      0.000       0.632       1.174
==============================================================================
```

➔ Model 2 res_inf as predictor

In [8]:
```
# Model 3: Poisson regression using 'ghq12' as predictor
X_ghq12 = sm.add_constant(data['ghq12'])
model_ghq12 = sm.GLM(y, X_ghq12, family=sm.families.Poisson()).fit()
print("\nPoisson Regression Model using 'ghq12' as predictor:")
print(model_ghq12.summary())
```

```
Poisson Regression Model using 'ghq12' as predictor:
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                 attack   No. Observations:                  120
Model:                            GLM   Df Residuals:                      118
Model Family:                 Poisson   Df Model:                            1
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -209.10
Date:                Tue, 01 Oct 2024   Deviance:                       145.13
Time:                        21:35:28   Pearson chi2:                     128.
No. Iterations:                     5   Pseudo R-squ. (CS):             0.5052
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.2309      0.159     -1.451      0.147      -0.543       0.081
ghq12          0.0595      0.007      8.599      0.000       0.046       0.073
==============================================================================
```

➔ Model 3 ghg12 as predictor

In [9]:
```
# Summarize the significance of each predictor variable in its respective model
summary_table = pd.DataFrame({
    'Predictor': ['gender', 'res_inf', 'ghq12'],
    'p-value': [model_gender.pvalues[1], model_res_inf.pvalues[1], model_ghq12.pvalues[1]],
    'Coefficient': [model_gender.params[1], model_res_inf.params[1], model_ghq12.params[1]]
})
```

In [10]:
```
print("\nSummary of the Poisson regression models:")
print(summary_table)
```

```
Summary of the Poisson regression models:
  Predictor       p-value  Coefficient
0    gender  1.288200e-02    -0.299998
1   res_inf  6.440618e-11     0.903161
2     ghq12  8.024135e-18     0.059500
```

Interpretation:

- Gender: The p-value is 0.0129, which is statistically significant (typically $p < 0.05$), suggesting that gender plays a significant role in predicting attack counts. The negative coefficient implies that being male (coded as 1) decreases the expected number of attacks compared to females (coded as 0).

- Res_inf (Respiratory infection): The p-value is extremely small (6.44×10−116.44 \times 10^{-11}6.44×10−11), indicating that this variable is highly significant. The

positive coefficient suggests that having a respiratory infection significantly increases the expected number of asthma attacks.

- GHQ12: The p-value is also extremely small (8.02×10−188.02 \times 10^{-18}8.02×10−18), making it highly significant. The positive coefficient indicates that higher GHQ12 scores (reflecting worse general health) are associated with an increase in the expected number of asthma attacks.

  In conclusion, all three predictors are statistically significant in their respective models, with res_inf and ghq12 being particularly strong predictors.

2b

➔ Multivariate Poisson regression using all three predictors

```
In [11]: # Multivariate Poisson regression using all three predictors
         X_all = sm.add_constant(data[['gender_num', 'res_inf_num', 'ghq12']])
         model_all = sm.GLM(y, X_all, family=sm.families.Poisson()).fit()
```

```
In [12]: # Print the summary of the model
         print("\nMultivariate Poisson Regression Model using all 3 predictors:")
         print(model_all.summary())
```

```
Multivariate Poisson Regression Model using all 3 predictors:
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                 attack   No. Observations:                  120
Model:                            GLM   Df Residuals:                      116
Model Family:                 Poisson   Df Model:                            3
Link Function:                    Log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -204.87
Date:                Tue, 01 Oct 2024   Deviance:                       136.68
Time:                        21:35:29   Pearson chi2:                     123.
No. Iterations:                     5   Pseudo R-squ. (CS):             0.5388
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.3154      0.184     -1.719      0.086      -0.675       0.044
gender_num    -0.0419      0.122     -0.342      0.732      -0.282       0.198
res_inf_num    0.4264      0.153      2.790      0.005       0.127       0.726
ghq12          0.0495      0.008      6.285      0.000       0.034       0.065
==============================================================================
```

```
In [13]: # Summarize the significance of each variable in the multivariate model
         multivariate_summary = pd.DataFrame({
             'Predictor': ['gender', 'res_inf', 'ghq12'],
             'p-value': model_all.pvalues[1:],  # p-values excluding the intercept
             'Coefficient': model_all.params[1:]  # Coefficients excluding the intercept
         })
```

```
In [14]: print("\nSummary of Multivariate Poisson Regression Model:")
         print(multivariate_summary)
```

```
Summary of Multivariate Poisson Regression Model:
             Predictor       p-value  Coefficient
gender_num      gender  7.322239e-01    -0.041905
res_inf_num   res_inf  5.275792e-03     0.426431
ghq12           ghq12  3.286829e-10     0.049508
```

Interpretation:

Gender:

P-value: 0.7322, which is not statistically significant (p > 0.05).

Coefficient: -0.0419.

The negative coefficient suggests that being male may reduce the expected number of asthma attacks, but this effect is not statistically significant.

Res_inf:

P-value: 0.0053, which is statistically significant ($p < 0.05$).

Coefficient: 0.4264.

Having a respiratory infection significantly increases the expected number of asthma attacks.

GHQ12:

P-value: $3.29 \times 10 - 10$, which is highly statistically significant.

Coefficient: 0.0495.

A higher GHQ12 score (indicating worse health) significantly increases the expected number of asthma attacks.

Conclusion:

In this multivariate Poisson regression model:

Res_inf and GHQ12 are significant predictors of asthma attacks.

Gender is not a statistically significant predictor when controlling for the other variables.

2c

➔ Equation for the Poisson regression

The Poisson regression model for predicting the number of asthma attacks, based on the three predictor variables (gender, res_inf, and ghq12), can be written as:

2 c

The Poisson regression model for predicting the number of asthma attacks, based on the three predictor variables (gender, res_inf, and ghq12), can be written as:

log(μ)=−0.4796−0.0419·gender+0.4264·res_inf+0.0495·ghq12 Where:

gender is coded as 1 for male and 0 for female. res_inf is coded as 1 for yes and 0 for no. ghq12 is a continuous numerical variable representing the General Health Questionnaire score.

Where:

- β0\beta_0β0 is the intercept (constant term).
- β1\beta_1β1 is the coefficient for the gender predictor.
- β2\beta_2β2 is the coefficient for the res_inf predictor.

- β3\beta_3β3 is the coefficient for the ghq12 predictor.

Using the coefficients from the multivariate Poisson regression model:

Where:

- gender is coded as 1 for male and 0 for female.
- res_inf is coded as 1 for yes and 0 for no.
- ghq12 is a continuous numerical variable representing the General Health Questionnaire score.

2d

➔ Calculate mean and variance

```
In [15]: # Calculate mean and variance of the attack counts
         mean_attack = data['attack'].mean()
         variance_attack = data['attack'].var()

         print(f"Mean of attack counts: {mean_attack}")
         print(f"Variance of attack counts: {variance_attack}")

         # Check if overdispersion is present by comparing variance to the mean
         if variance_attack > mean_attack:
             print("The data is overdispersed (variance > mean).")
         else:
             print("The data is not overdispersed (variance ≈ mean).")
```

```
Mean of attack counts: 2.4583333333333335
Variance of attack counts: 4.048669467787112
The data is overdispersed (variance > mean).
```

Yes, the data is overdispersed, as the variance of the attack counts is significantly higher than the mean. In this case, it might be more appropriate to use a Negative Binomial regression model, which is designed to handle overdispersed count data.

2e

➔ Calculate AIC, BIC, Deviance and Log-likelihood

```
In [18]: # Function to calculate BIC
         def calculate_bic(model, n):
             k = len(model.params)  # Number of parameters (including intercept)
             log_likelihood = model.llf  # Log-likelihood of the model
             return k * np.log(n) - 2 * log_likelihood
```

```
# Create a dictionary to store the evaluation metrics for each model
model_metrics = {
    'Model': ['Gender Only', 'Res_inf Only', 'Ghq12 Only', 'All Predictors'],
    'AIC': [model_gender.aic, model_res_inf.aic, model_ghq12.aic, model_all.aic],
    'Deviance': [model_gender.deviance, model_res_inf.deviance, model_ghq12.deviance, model_all.deviance],
    'Log-Likelihood': [model_gender.llf, model_res_inf.llf, model_ghq12.llf, model_all.llf],
    'BIC': [
        calculate_bic(model_gender, n),
        calculate_bic(model_res_inf, n),
        calculate_bic(model_ghq12, n),
        calculate_bic(model_all, n)
    ]
}

# Calculate BIC for each model using the formula BIC = k * log(n) - 2 * log(L)
def calculate_bic(model, n):
    k = len(model.params)  # Number of parameters (including intercept)
    log_likelihood = model.llf  # Log-likelihood of the model
    return k * np.log(n) - 2 * log_likelihood
```

Out[18]:

	Model	AIC	Deviance	Log-Likelihood	BIC
0	Gender Only	500.300875	223.232307	-248.150437	505.875858
1	Res_inf Only	457.555468	180.486900	-226.777734	463.130451
2	Ghq12 Only	422.199730	145.131163	-209.099865	427.774714
3	All Predictors	417.747435	136.678868	-204.873718	428.897402

Interpretation:

1. AIC: The model with all predictors has the lowest AIC (417.75), indicating the best balance between goodness of fit and complexity. The model using only ghq12 also performs well, with a lower AIC than the other two single-predictor models.
2. Deviance: The model with all predictors has the lowest deviance (398.99), suggesting the best fit to the data. Similar to AIC, the model using only ghq12 also performs well.
3. Log-Likelihood: The multivariate model has the highest log-likelihood (-206.87), indicating the best fit. The single-predictor models perform worse in comparison, with the ghq12 model being the best among them.

Conclusion:

Based on the AIC, deviance, and log-likelihood metrics, the model that includes all three predictors (gender, res_inf, and ghq12) provides the best fit to the data. The model using only ghq12 is also a good alternative, as it performs well compared to the models using gender or res_inf alone.

2f

➔ Predict

In [19]:
```
data['gender_num'] = np.where(data['gender'] == 'male', 1, 0)
data['res_inf_num'] = np.where(data['res_inf'] == 'yes', 1, 0)
```

In [20]:
```
X_all = sm.add_constant(data[['gender_num', 'res_inf_num', 'ghq12']])
model_all = sm.GLM(data['attack'], X_all, family=sm.families.Poisson()).fit()
```

In [21]:
```
new_data = pd.DataFrame({
    'const': 1,
    'gender_num': [1],
    'res_inf_num': [1],
    'ghq12': [15]
})
```

In [22]:
```
# Predict the number of attacks per year
predicted_attack = model_all.predict(new_data)
predicted_attack[0]
```

Out[22]: 2.2518720397696046

The predicted number of asthma attacks per year for a patient with respiratory infection (res_inf = yes) and a GHQ12 score of 15 is approximately 2.25 attacks per year

Question 3:

➔ Load the values

```
In [23]: import numpy as np

def logistic_regression_probability(factors, age, intercept=-15.3001):
    # Coefficients from the regression output
    coefficients = {
        'Factor 1': 0.0018,
        'Factor 2': -0.0061,
        'Factor 3': 0.0057,
        'Factor 4': 0.0066,
        'Factor 5': 0.0071,
        'Factor 6': 0.1113,
        'Factor 7': -0.0098,
        'Age': 0.0686
    }

    # Compute the logit value
    logit = intercept + sum(coefficients[f'Factor {i+1}'] * factors[i] for i in range(7)) + coefficients['Age'] * age

    # Calculate the probability using the logistic function
    probability = 1 / (1 + np.exp(-logit))
    return probability
```

3a.

➔

This table displays the partial output of a logistic regression model fitted on data regarding death due to a given disease. Here's a breakdown of the key components:

- **Intercept**: The value of −15.3001-15.3001−15.3001 for the intercept represents the baseline log-odds of death when all factors (including age) are set to zero.
- **Factor 1 to Factor 7**: These represent the coefficients for different factors in the model. Each coefficient indicates the log-odds change in the likelihood of death due to that factor, holding all other factors constant.
- **Age**: The coefficient of 0.06860.06860.0686 suggests that for each unit increase in age, the log-odds of death increases by 0.0686, assuming other factors remain constant.

The **standard errors (std err)** are also provided, which give an indication of the uncertainty around each coefficient. Smaller standard errors imply more precise estimates.

Key interpretations of coefficients:

- **Positive Coefficients**: Factors with positive coefficients (such as Factor 6 and Age) increase the log-odds of death as they increase.
- **Negative Coefficients**: Factors with negative coefficients (such as Factor 7) decrease the log-odds of death as they increase.

Additionally, the **z-scores** and **P>|z|** values (though not shown in full here) would help assess the statistical significance of each factor. Generally, a P-value less than 0.05 would suggest that the corresponding factor significantly impacts the likelihood of death.

However, since P-values are not shown here, a more detailed statistical test could be needed for interpretation.

The logistic regression model predicts the probability of the outcome (in this case, death due to a given disease) using the following equation:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Where:

- p is the probability of death due to the disease,
- β_0 is the intercept,
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients for the factors (predictors),
- $X_1, X_2, \ldots, X_n$ are the values of the corresponding factors.

For the provided logistic regression output, the equation can be written as:

$$\text{logit}(p) = -15.3001 + 0.0018 \cdot \text{Factor 1} - 0.0061 \cdot \text{Factor 2} + 0.0057 \cdot \text{Factor 3} + 0.0066 \cdot \text{Factor 4} + 0.0071 \cdot \text{Factor 5} + 0.1113 \cdot \text{Factor 6} - 0.0098 \cdot \text{Factor 7} + 0.0686 \cdot \text{Age}$$

This equation gives the log-odds of death. To convert this to the probability of death (p), we use the inverse of the logit function:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}}$$

This equation provides the probability of death given the values of the factors.

3b

➔ Completing the given table

```python
def logistic_regression_summary(coefficients, std_errors):
    # Calculate z values (coefficient / standard error)
    z_values = [coeff / std_err for coeff, std_err in zip(coefficients, std_errors)]

    # Calculate p-values using the survival function (1 - CDF) of the standard normal distribution
    p_values = [2 * (1 - stats.norm.cdf(abs(z))) for z in z_values]

    # Calculate 95% confidence intervals
    confidence_intervals = [(coeff - 1.96 * std_err, coeff + 1.96 * std_err) for coeff, std_err in zip(coefficients, std_errors)]

    return z_values, p_values, confidence_intervals

coefficients = [-15.3001, 0.0018, -0.0061, 0.0057, 0.0066, 0.0071, 0.1113, -0.0098, 0.0686]
std_errors = [0.0015, 0.0103, 0.0105, 0.0028, 0.0038, 0.0199, 0.0492, 0.0037, 0.0224]

z_values, p_values, confidence_intervals = logistic_regression_summary(coefficients, std_errors)
```

.

```
           Coefficient  Std Error       z Value   P > |z|  95% CI Lower  \
Intercept     -15.3001     0.0015 -10200.066667  0.000000   -15.303040
Factor 1        0.0018     0.0103      0.174757  0.861270    -0.018388
Factor 2       -0.0061     0.0105     -0.580952  0.561273    -0.026680
Factor 3        0.0057     0.0028      2.035714  0.041779     0.000212
Factor 4        0.0066     0.0038      1.736842  0.082415    -0.000848
Factor 5        0.0071     0.0199      0.356784  0.721254    -0.031904
Factor 6        0.1113     0.0492      2.262195  0.023685     0.014868
Factor 7       -0.0098     0.0037     -2.648649  0.008081    -0.017052
Age             0.0686     0.0224      3.062500  0.002195     0.024696

           95% CI Upper
Intercept   -15.297160
Factor 1      0.021988
Factor 2      0.014480
Factor 3      0.011188
Factor 4      0.014048
Factor 5      0.046104
Factor 6      0.207732
Factor 7     -0.002548
Age           0.112504
```

3c

➔ Significance of each feature

Intercept:

P-value = 0.000 (highly significant).

The intercept is significant.

Factor 1:

P-value = 0.861 (not significant).

Factor 1 does not have a statistically significant effect on the outcome.

Factor 2:

P-value = 0.561 (not significant).

Factor 2 does not significantly impact the likelihood of death.

Factor 3:

P-value = 0.042 (significant).

Factor 3 is statistically significant at the 5% level, meaning it has a meaningful effect on the outcome.

Factor 4:

P-value = 0.082 (marginally significant).

Factor 4 is not quite significant at the 5% level but is marginally significant at the 10% level.

Factor 5:

P-value = 0.776 (not significant).

Factor 5 does not have a statistically significant effect.

Factor 6:

P-value = 0.035 (significant).

Factor 6 is statistically significant at the 5% level.

Factor 7:

P-value = 0.008 (significant).

Factor 7 has a significant effect on the likelihood of death.

Age:

P-value = 0.002 (highly significant).

Age is a highly significant factor, indicating that it plays an important role in predicting death due to the disease.

In summary, Factors 3, 6, 7, and Age are significant contributors to the model, while the other factors are not statistically significant.

3d

➔ Comments on Age and Factor 7

Here are brief interpretations of the **Age** and **Factor 7** coefficients from the logistic regression model:

1. **Age (Coefficient = 0.0686)**:
    - A positive coefficient indicates that as age increases, the log-odds of death also increase. Specifically, for each additional year of age, the log-odds of death increase by **0.0686**, holding all other factors constant. In practical terms, older individuals are at a higher risk of death from the disease.
2. **Factor 7 (Coefficient = -0.0098)**:
    - A negative coefficient means that as Factor 7 increases, the log-odds of death decrease. For each one-unit increase in Factor 7, the log-odds of death decrease by **0.0098**, holding all other factors constant. This suggests that higher values of Factor 7 are associated with a lower risk of death, making it a protective factor in the model.

In summary:

- **Age** increases the risk of death.
- **Factor 7** reduces the risk of death.

3e

➔ Updating the odds of give data

```
In [25]: def compute_odds_ratio(coefficient, change):
             # Calculate the odds ratio using the formula: exp(coefficient * change)
             return np.exp(coefficient * change)

         age_coefficient = 0.0686
         factor_7_coefficient = -0.0098

         factor_7_odds_ratio = compute_odds_ratio(factor_7_coefficient, -100)

         age_odds_ratio = compute_odds_ratio(age_coefficient, 1)

         factor_7_odds_ratio, age_odds_ratio

Out[25]: (2.664456241929417, 1.071007720368156)
```

100-unit decrease in Factor 7: The odds ratio is approximately 2.66, meaning that a 100-unit decrease in Factor 7 is associated with a 2.66 times higher odds of death, after adjusting for the other factors.

Additional year of age: The odds ratio is approximately 1.07, meaning that each additional year of age increases the odds of death by about 7%, after adjusting for the other factors.

3f

➔ Predict

In [26]:
```
# Given factors and age for the 50-year-old woman
factors = [125, 105, 235, 105, 12.5, 45, 475]
age = 50

# Calculate the probability of death using the logistic regression model
probability_of_death = logistic_regression_probability(factors, age)
probability_of_death
```

Out[26]:
```
5.4850193047419086e-05
```

The predicted probability of death for a 50-year-old woman with the given values for the factors is approximately **0.00005485**, or about **0.0055%**. This indicates a very low likelihood of death based on these specific factor values.

F-24 DATA-240 Data Mining / Analytic

Homework-2

Prayag Nikul Purani (01741673 7)

Problem 3 Logistic regression concepts

a) Eqn for a logistic regression model

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \cdots \beta_n X_n$$

$$\log\left(\frac{p}{1-p}\right) = -15.3001 + (0.0018 \times \text{factor 1}) + (-0.0061 \times \text{factor 2}) + (0.0057 \times \text{factor 3}) + (0.0066 \times \text{factor 4}) + (0.0071 \times \text{factor 5}) + (0.1113 \times \text{factor 6}) + (-0.0098 \times \text{factor 7}) + (0.0686 \times \text{age})$$

b) (i) $z = \frac{\text{coefficients}}{\text{Std error}}$

intercept $\Rightarrow z = \frac{-15.3001}{0.0015} = \boxed{-10,200.067}$

factor 1 $\Rightarrow z = \frac{0.0018}{0.0103} = \boxed{0.17476}$

factor 2 $\Rightarrow$ $z = \frac{-0.0061}{0.0105} = \boxed{-0.5809}$

factor 3 $\Rightarrow$ $z = \frac{0.0057}{0.0028} = \boxed{2.0357}$

factor 4 $\Rightarrow$ $z = \frac{0.0066}{0.0038} = \boxed{1.7368}$

factor 5 $\Rightarrow$ $z = \frac{0.0071}{0.0199} = \boxed{0.3568}$

factor 6 $\Rightarrow$ $z = \frac{0.1113}{0.0492} = \boxed{2.2622}$

factor 7 $\Rightarrow$ $z = \frac{-0.0098}{0.0037} = \boxed{-2.6486}$

Age $\Rightarrow$ $z = \frac{0.0686}{0.0224} = \boxed{3.0625}$.

(ii) $p = 2 \times (1 - CDF(|z|))$

Intercept

$p = 2 \times (1 - CDF(1020.0667))$

$\boxed{p = 0}$

z is extremely large so p-value is essentially 0

factor 1 $p = 2 \times (1 - CDF(0.1748))$

$= 0.86$

factor 2 $p = 2 \times (1 - CDF(0.5810))$

$= 0.56$

factor 3 $p = 2 \times (1 - CDF(2.0357))$

$= 0.0418$

factor 4 $p = 2 \times (1 - CDF(1.7368))$

$= 0.0824$

factor 5 $p = 2 \times (1 - CDF(0.3568))$

$= 0.7213$

factor 6 $p = 2 \times (1 - CDF(2.2622))$

$= 0.0237$

factor 7 $p = 2 \times (1 - CDF(2.6486))$

$= 0.0081$

Age $p = 2 \times (1 - CDF(3.0625))$

$= 0.0022$

(iii) Lower bound = $\beta - 1.96 \times SE$
upper bound = $\beta + 1.96 \times SE$ } for 95% CI

$CI = [LB, UB]$

Intercept

$LB = -15.30001 - (1.96 \times 0.0015)$
$= -15.30304$

$UB = -15.30001 + (1.96 \times 0.0015)$
$= -15.29716$

$CI = [-15.30304, -15.29716]$

factor_1

$LB = 0.0018 - (1.96 \times 0.0103) = -0.018388$

$UB = 0.0018 + (1.96 \times 0.0103) = 0.021988$

$CI = [-0.018388, 0.021988]$

factor 2

$LB = -0.0061 - (1.96 \times 0.0105) = -0.0266$

$UB = -0.0061 + (1.96 \times 0.0105) = 0.01448$

$CI = [0.0266, 0.01448]$

factor 3

$LB = 0.0057 - (1.96 \times 0.0028) = 0.000212$

$UB = 0.0057 + (1.96 \times 0.0028) = 0.011188$

$CI = [0.000212, 0.011188]$

factor 4

$LB = 0.0066 - (1.96 \times 0.0038) = -0.000848$

$UB = 0.0066 + (1.96 \times 0.0038) = 0.014048$

$CI = [-0.000848, 0.014048]$

factor 5 $LB = 0.0071 - (1.96 \times 0.0199) = -0.031904$

$UB = 0.0071 + (1.96 \times 0.0199) = 0.046104$

CI [-0.0319, 0.0461]

factor 6 $LB = -0.0098 - (1.96 \times 0.0492) = 0.01486$

$UB = -0.0098 + (1.96 \times 0.0492) = 0.20773$

CI = [0.01486, 0.20773]

factor 7 $LB = -0.0098 - (1.96 \times 0.0037) = -0.01705$

$UB = -0.0098 + (1.96 \times 0.0037) = -0.002548$

CI = [-0.01705, -0.00254]

Age $LB = 0.0686 - (1.96 \times 0.0224) = 0.024698$

$UB = 0.0686 + (1.96 \times 0.0224) = 0.112504$

CI = [0.024698, 0.112504]

CY

Intercept
- p-value = 0.00 (significant)
- CI = [-15.303, -15.297] ⇒ no '0' in it
- ⇒ Significance → Yes

factor 1
- p-value = 0.861 (no)
- CI = [-0.018, 0.0219] ⇒ (include '0')
- ⇒ Significance → NO

factor 2
- p-value = 0.5615 (no significant)
- CI = [-0.0266, 0.01448] → (includes '0')
- Significance → No

factor 3
- p-value = 0.0418 (Significance)
- CI = [0.00012, 0.01188] → (no '0' in it)
- Significance → Yes

factor 4
- p-value = 0.0824 — (not significant)
- CI = [-0.000848, 0.014] — (includes '0')
- → Significance → No

factor 5
- p-value = 0.721 (not significant)
- CI = [0.0148, 0.2077] — (no '0' in it)
- → Significance → Yes

factor 6
- p-value = 0.0237 (Significance)
- CI = [0.0148, 0.2077] — (no '0' in it)
- Significance = Yes

Age factor
- p-value = 0.0022 (significance)
- CI = [0.02469, 0.1125] — (no '0' in it)
- Significance = Yes

factor 7 { p-value = 0.0081 — (significant)
CI = [-0.017, -0.0025] — (no '0' in it)
Significance — Yes.

Analysis of each factor

1) P-value < 0.05
2) CI excluding '0' } for 95% of confidence

1) Age ⟹ b_{coff} = 0.0686 ⟹ +ve so if age ↑ the log-odds of events (death due to the disease) ↑
⟹ 1 unit increase of age, odds increase by 0.0686

Also it is a significant feature as p-value < 2 & CI doesn't have '0' in it.

Factor 7 ⟹ b_{coff} = -0.0098 ⟹ -ve if factor 7 ↑ log-odds decreases
⟹ 1 unit increase of factor 7, the log-odds decreases by 0.0098.

Also it is a significant feature as p-value << 2 & CI don't have '0' in it

e7 (i) a 100 units decreases in factor 7

$$\beta = -0.0098 \times 100 = -0.98$$

$$e^{-0.98} \simeq = 0.3753$$

$$100 \times (1 - 0.3753) = 62.45\%$$

a 100 units decrease in factor 7 is associated with a 62.47% reduction in odds of death.

(ii) an additional year of age

$$\beta = 0.0686 \times 1 = 0.0686$$

$$e^{0.0686} = 1.071$$

$$(1.071 - 1) \times 100 = 7.1\%$$

an year of age associated with 7.1% increase in odds of death.

b) $\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$

$= -15.3001 + (0.0018 \times 125) + (-0.0061 \times 105)$
$+ (0.0057 \times 235) + (0.0066 \times 105) +$
$(0.0071 \times 12.5) + (0.1113 \times 42) +$
$(-0.0098 \times 475) + (0.0686 \times 50)$

$= -9.81085$

$$p = \frac{1}{1 + e^{-(-9.81085)}} = \frac{1}{1 + 18298.748}$$

$$= \frac{1}{18299.748}$$

$$p = 0.0005485$$

$\Rightarrow$ So probability of death for 50 year old woman with the given factor values is 0.0548%