1.4.2 How can we estimate the likelihood of experiencing high stress levels given an individual's technology usage patterns and work environment impact?

```
from google.colab import files
```

```
uploaded = files.upload()
```

```
Choose Files  No file chosen        Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving mental_health_and_technology_usage_2022.xlsx to mental_health_and_technology_usage_2022.xlsx
```

```
import pandas as pd
data = pd.read_excel('mental_health_and_technology_usage_2022.xlsx')
```

```
print(data.columns)
```

```
Index(['User_ID', 'Age', 'Birth Year', 'Generation', 'Technology_Usage_Hours',
       'Social_Media_Usage_Hours', 'Gaming_Hours', 'Screen_Time_Hours',
       'Mental_Health_Status', 'Stress_Level', 'Sleep_Hours',
       'Physical_Activity_Hours', 'Support_Systems_Access',
       'Work_Environment_Impact', 'Online_Support_Usage'],
      dtype='object')
```

```
print(data)
```

```
         User_ID  Age  Birth Year   Generation  Technology_Usage_Hours  \
0     USER-00001   23        1999        Gen Z                    6.57
1     USER-00002   21        2001        Gen Z                    3.01
2     USER-00003   51        1971        Gen X                    3.04
3     USER-00004   25        1997        Gen Z                    3.84
4     USER-00005   53        1969        Gen X                    1.20
...          ...  ...         ...          ...                     ...
9995  USER-09996   42        1980        Gen X                    7.05
9996  USER-09997   31        1991  Millennials                    3.12
9997  USER-09998   23        1999        Gen Z                    4.38
9998  USER-09999   38        1984  Millennials                    4.44
9999  USER-10000   41        1981  Millennials                    2.50

      Social_Media_Usage_Hours  Gaming_Hours  Screen_Time_Hours  \
0                         6.00          0.68              12.36
1                         2.57          3.74               7.61
2                         6.14          1.26               3.16
3                         4.48          2.59              13.08
4                         0.56          0.29              12.63
...                        ...           ...                ...
9995                      0.41          0.53              13.90
9996                      6.79          0.80               1.17
9997                      3.98          0.52               7.81
9998                      1.48          3.28              13.95
9999                      4.80          0.25               8.82

     Mental_Health_Status Stress_Level  Sleep_Hours  Physical_Activity_Hours  \
0                    Good          Low         8.01                     6.71
1                    Poor         High         7.28                     5.88
2                    Fair         High         8.04                     9.81
3               Excellent       Medium         5.62                     5.28
4                    Good          Low         5.55                     4.00
...                   ...          ...          ...                      ...
9995                 Good       Medium         7.37                     5.02
9996                 Fair       Medium         8.92                     9.78
9997                 Poor         High         7.59                     2.99
9998                 Poor       Medium         7.26                     2.24
9999                 Fair          Low         4.62                     5.09

     Support_Systems_Access Work_Environment_Impact Online_Support_Usage
0                        No                Negative                  Yes
1                       Yes                Positive                   No
2                        No                Negative                   No
3                       Yes                Negative                  Yes
4                        No                Positive                  Yes
...                     ...                     ...                  ...
9995                    Yes                 Neutral                   No
9996                     No                 Neutral                  Yes
9997                     No                Positive                   No
9998                    Yes                 Neutral                  Yes
9999                    Yes                Positive                  Yes

[10000 rows x 15 columns]
```

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report



# Check column names to ensure they are correct
print(data.columns)

# If there are extra spaces in the column names, remove them
data.columns = data.columns.str.strip()

# Preprocessing: Convert Stress_Level to binary (1 for 'High', 0 for 'Low' or 'Medium')
data['Stress_Level_Binary'] = data['Stress_Level'].apply(lambda x: 1 if x == 'High' else 0)

# Encoding categorical variables
le = LabelEncoder()
data['Mental_Health_Status'] = le.fit_transform(data['Mental_Health_Status'])
data['Support_Systems_Access'] = le.fit_transform(data['Support_Systems_Access'])
data['Online_Support_Usage'] = le.fit_transform(data['Online_Support_Usage'])
data['Work_Environment_Impact'] = le.fit_transform(data['Work_Environment_Impact'])

# Select independent variables (Technology usage, work environment impact, etc.)
X = data[['Technology_Usage_Hours', 'Social_Media_Usage_Hours', 'Screen_Time_Hours',
          'Work_Environment_Impact', 'Physical_Activity_Hours', 'Sleep_Hours']]
y = data['Stress_Level_Binary']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Logistic Regression model
lr_model = LogisticRegression(max_iter=1000)  # Increased max_iter to ensure convergence
lr_model.fit(X_train, y_train)

# Predictions and evaluation
y_pred = lr_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

# Output the results
print("Accuracy: ", accuracy)
print("Classification Report: \n", report)
```

```
Index(['User_ID', 'Age', 'Birth Year', 'Generation', 'Technology_Usage_Hours',
       'Social_Media_Usage_Hours', 'Gaming_Hours', 'Screen_Time_Hours',
       'Mental_Health_Status', 'Stress_Level', 'Sleep_Hours',
       'Physical_Activity_Hours', 'Support_Systems_Access',
       'Work_Environment_Impact', 'Online_Support_Usage'],
      dtype='object')
Accuracy:  0.6813333333333333
Classification Report: 
               precision    recall  f1-score   support

           0       0.68      1.00      0.81      2044
           1       0.00      0.00      0.00       956

    accuracy                           0.68      3000
   macro avg       0.34      0.50      0.41      3000
weighted avg       0.46      0.68      0.55      3000

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```