Lab 10: MLP Classifier

Created by: Preksha Shah | 2348446

Date: 18.04.2024

Basic EDA

```
# Importing necessary libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.decomposition import PCA
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MaxAbsScaler
```

Explanation:

- **pandas (pd):** Used for data manipulation and analysis.
- **numpy (np):** Provides support for mathematical operations on arrays and matrices.
- **matplotlib.pyplot (plt):** Used for creating visualizations such as plots and charts.
- **seaborn (sns):** Built on top of matplotlib, seaborn provides enhanced visualizations and statistical graphics.
- **sklearn.model_selection.train_test_split:** Used to split the dataset into training and testing sets.
- **sklearn.preprocessing.StandardScaler:** Used for standardization or normalization of features.
- **sklearn.neural_network.MLPClassifier:** Implements a Multi-layer Perceptron classifier, which will be used for the neural network model.
- **sklearn.metrics:** Provides various metrics for evaluating model performance, such as accuracy, precision, recall, etc.
- **sklearn.decomposition.PCA:** Performs Principal Component Analysis, which will be used for dimensionality reduction.

```
# Load the dataset into Python environment
data = pd.read_csv('/content/survey.csv')
```

```
# Display basic information about the dataset
print("Basic Information About the Dataset:")
print(data.info())
```

```
Basic Information About the Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Timestamp                  1259 non-null   object
 1   Age                        1259 non-null   int64
 2   Gender                     1259 non-null   object
 3   Country                    1259 non-null   object
 4   state                      744 non-null    object
 5   self_employed              1241 non-null   object
 6   family_history             1259 non-null   object
 7   treatment                  1259 non-null   object
 8   work_interfere             995 non-null    object
 9   no_employees               1259 non-null   object
 10  remote_work                1259 non-null   object
 11  tech_company               1259 non-null   object
 12  benefits                   1259 non-null   object
 13  care_options               1259 non-null   object
 14  wellness_program           1259 non-null   object
 15  seek_help                  1259 non-null   object
 16  anonymity                  1259 non-null   object
 17  leave                      1259 non-null   object
 18  mental_health_consequence  1259 non-null   object
 19  phys_health_consequence    1259 non-null   object
 20  coworkers                  1259 non-null   object
 21  supervisor                 1259 non-null   object
 22  mental_health_interview    1259 non-null   object
```

```
 23  phys_health_interview      1259 non-null   object
 24  mental_vs_physical         1259 non-null   object
 25  obs_consequence            1259 non-null   object
 26  comments                   164 non-null    object
dtypes: int64(1), object(26)
memory usage: 265.7+ KB
None
```

Observations:

- **RangeIndex:** Indicates that the DataFrame has 1259 entries (rows), indexed from 0 to 1258.
- **Data columns:** Total 27 columns are present in the dataset.
- **Column Information:**
    1. **Timestamp:** Contains timestamps of when the survey was conducted.
    2. **Age:** Represents the age of respondents (numerical variable).
    3. **Gender:** Indicates the gender of respondents.
    4. **Country:** Represents the country of respondents.
    5. **State:** Indicates the state or territory where respondents from the United States live.
    6. **Self-employed:** Indicates whether respondents are self-employed.
    7. **Family history:** Indicates if respondents have a family history of mental illness.
    8. **Treatment:** Indicates if respondents have sought treatment for a mental health condition.
    9. **Work interference:** Indicates if respondents feel that their mental health condition interferes with their work.
    10. **No. of employees:** Represents the size of the company or organization where respondents work.
    11. **Remote work:** Indicates if respondents work remotely at least 50% of the time.
    12. **Tech company:** Indicates if the employer is primarily a tech company/organization.
    13. **Benefits:** Indicates if the employer provides mental health benefits.
    14. **Care options:** Indicates if respondents know the options for mental health care provided by their employer.
    15. **Wellness program:** Indicates if respondents' employer has discussed mental health as part of an employee wellness program.
    16. **Seek help:** Indicates if respondents' employer provides resources to learn more about mental health issues and how to seek help.
    17. **Anonymity:** Indicates if respondents' anonymity is protected if they choose to take advantage of mental health or substance abuse treatment resources.
    18. **Leave:** Indicates the ease of taking medical leave for a mental health condition.
    19. **Mental health consequence:** Indicates if discussing a mental health issue with the employer would have negative consequences.
    20. **Physical health consequence:** Indicates if discussing a physical health issue with the employer would have negative consequences.
    21. **Coworkers:** Indicates if respondents would be willing to discuss a mental health issue with their coworkers.
    22. **Supervisor:** Indicates if respondents would be willing to discuss a mental health issue with their direct supervisor(s).
    23. **Mental health interview:** Indicates if respondents would bring up a mental health issue with a potential employer in an interview.
    24. **Physical health interview:** Indicates if respondents would bring up a physical health issue with a potential employer in an interview.
    25. **Mental vs physical:** Indicates if respondents feel that their employer takes mental health as seriously as physical health.
    26. **Obs consequence:** Indicates if respondents have heard of or observed negative consequences for coworkers with mental health conditions in their workplace.
    27. **Comments:** Additional notes or comments provided by respondents.
- **Non-Null Count:** Indicates the number of non-null values present in each column.
- **Dtype:** Indicates the data type of each column.
- **Memory Usage:** Indicates the memory usage of the DataFrame.

This information gives us an overview of the dataset's structure, including the number of samples, features, data types, and any missing values present.

```
# Display the first few rows of the dataset
print("\nFirst Few Rows of the Dataset:")
print(data.head())
```

```
First Few Rows of the Dataset:
             Timestamp  Age  Gender         Country state self_employed  \
0  2014-08-27 11:29:31   37  Female   United States    IL           NaN
1  2014-08-27 11:29:37   44       M   United States    IN           NaN
2  2014-08-27 11:29:44   32    Male          Canada   NaN           NaN
3  2014-08-27 11:29:46   31    Male  United Kingdom   NaN           NaN
4  2014-08-27 11:30:22   31    Male   United States    TX           NaN

  family_history treatment work_interfere    no_employees  ...  \
0             No       Yes          Often            6-25  ...
1             No        No         Rarely  More than 1000  ...
2             No        No         Rarely            6-25  ...
3            Yes       Yes          Often          26-100  ...
```

```
4             No        No          Never         100-500  ...

                leave mental_health_consequence phys_health_consequence  \
0       Somewhat easy                        No                      No
1          Don't know                     Maybe                      No
2  Somewhat difficult                        No                      No
3  Somewhat difficult                       Yes                     Yes
4          Don't know                        No                      No

     coworkers supervisor mental_health_interview phys_health_interview  \
0  Some of them        Yes                      No                 Maybe
1           No         No                      No                    No
2          Yes        Yes                     Yes                   Yes
3  Some of them         No                   Maybe                 Maybe
4  Some of them        Yes                     Yes                   Yes

  mental_vs_physical obs_consequence comments
0                Yes              No      NaN
1         Don't know              No      NaN
2                 No              No      NaN
3                 No             Yes      NaN
4         Don't know              No      NaN

[5 rows x 27 columns]
```

```
#Check the dataset for missing data
if data.isnull().sum().sum() == 0 :
    print ('There is no missing data in our dataset')
else:
    print('There is {} missing data in our dataset '.format(data.isnull().sum().sum()))
```

```
There is 1892 missing data in our dataset
```

```
#Check our missing data from which columns and how many unique features they have.
frame = pd.concat([data.isnull().sum(), data.nunique(), data.dtypes], axis = 1, sort= False)
frame
```

	0	1	2
Timestamp	0	1246	object
Age	0	53	int64
Gender	0	49	object
Country	0	48	object
state	515	45	object
self_employed	18	2	object
family_history	0	2	object
treatment	0	2	object
work_interfere	264	4	object
no_employees	0	6	object
remote_work	0	2	object
tech_company	0	2	object
benefits	0	3	object
care_options	0	3	object
wellness_program	0	3	object
seek_help	0	3	object
anonymity	0	3	object
leave	0	5	object
mental_health_consequence	0	3	object
phys_health_consequence	0	3	object
coworkers	0	3	object
supervisor	0	3	object
mental_health_interview	0	3	object
phys_health_interview	0	3	object
mental_vs_physical	0	3	object
obs_consequence	0	2	object
comments	1095	160	object

Next steps: View recommended plots

- Four columns have missing data, state , work_interfere, self_employed and comments.
- State and comments are not important to me, so I'm gonna drop them but, we need to fill in Missing data for work_interfere and, self_employed

```
# Drop unnecessary columns
columns_to_drop = ['state', 'comments', 'Timestamp']
data = data.drop(columns=columns_to_drop)


#Fill in missing values in specific columns
data['work_interfere'] = SimpleImputer(strategy='most_frequent').fit_transform(data['work_interfere'].values.reshape(-1, 1)).ravel()
data['self_employed'] = SimpleImputer(strategy='most_frequent').fit_transform(data['self_employed'].values.reshape(-1, 1)).ravel()


#Clean and organize data in the 'Gender' column
data['Gender'].replace(['Male ', 'male', 'M', 'm', 'Male', 'Cis Male',
                        'Man', 'cis male', 'Mail', 'Male-ish', 'Male (CIS)',
                        'Cis Man', 'msle', 'Malr', 'Mal', 'maile', 'Make'], 'Male', inplace=True)

data['Gender'].replace(['Female ', 'female', 'F', 'f', 'Woman', 'Female',
                        'femail', 'Cis Female', 'cis-female/femme', 'Femake', 'Female (cis)',
                        'woman'], 'Female', inplace=True)

data["Gender"].replace(['Female (trans)', 'queer/she/they', 'non-binary', 'fluid', 'queer', 'Androgyne', 'Trans-female', 'male leaning a
                       'Agender', 'A little about you', 'Nah', 'All',
                       'ostensibly male, unsure what that really means',
                       'Genderqueer', 'Enby', 'p', 'Neuter', 'something kinda male?',
                       'Guy (-ish) ^_^', 'Trans woman'], 'Other', inplace=True)
```

```
# Check for duplicated data
if data.duplicated().sum() == 0:
    print('There is no duplicated data:')
else:
    print('There is {} duplicated data:'.format(data.duplicated().sum()))
    data.drop_duplicates(inplace=True)
```

```
There is 4 duplicated data:
```

```
#Filter and clean data in the 'Age' column
data.drop(data[data['Age'] < 0].index, inplace=True)
data.drop(data[data['Age'] > 99].index, inplace=True)
```

- **Removing Negative Values:** By dropping rows where the 'Age' column has values less than 0, the code eliminates any entries with negative ages. Negative ages are logically incorrect and likely represent data entry errors or anomalies. Removing them ensures that the dataset contains only valid age values.
- **Removing Unreasonably High Values:** Similarly, by dropping rows where the 'Age' column has values greater than 99, the code filters out any entries with unreasonably high ages. In many contexts, ages above 99 are considered outliers or data anomalies. Removing them helps prevent skewed analysis results and improves the overall quality of the dataset.

```
# Initialize LabelEncoder
le = LabelEncoder()

# Use LabelEncoder to change the data types to 'int
columns_to_encode = ['Gender', 'Country', 'self_employed','family_history', 'treatment', 'work_interfere','no_employees',
                     'remote_work', 'tech_company','benefits','care_options', 'wellness_program',
                     'seek_help', 'anonymity', 'leave', 'mental_health_consequence', 'phys_health_consequence',
                     'coworkers', 'supervisor', 'mental_health_interview','phys_health_interview',
                     'mental_vs_physical', 'obs_consequence']

for columns in columns_to_encode:
    data[columns] = le.fit_transform(data[columns])
```

The `LabelEncoder` is used to convert categorical variables into numerical representations, enabling compatibility with machine learning algorithms that require numerical input. This transformation improves model performance, simplifies data processing, and facilitates dimensionality reduction.

Univariate Analysis

```
# Identify numerical and categorical variables
numerical_variables = data.select_dtypes(include=['int', 'float']).columns.tolist()
categorical_variables = data.select_dtypes(include=['object']).columns.tolist()
```

For numerical variables

```
#Calculate basic descriptive statistics
print("Basic Descriptive Statistics for Numerical Variables:")
print(data.describe())
```

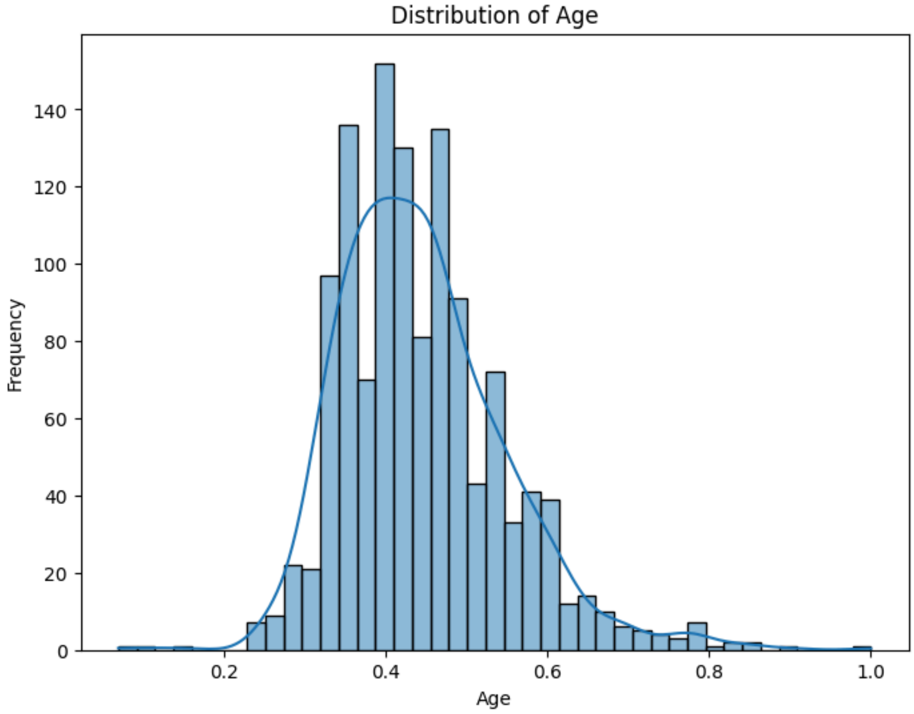
```
Basic Descriptive Statistics for Numerical Variables:
               Age      Gender       Country  self_employed  family_history  \
count  1250.000000  1250.00000  1.250000e+03    1250.000000     1250.000000
mean      0.444778     0.81760  3.410605e-17       0.114400        0.390400
std       0.102557     0.42388  1.000400e+00       0.318424        0.488035
min       0.069444     0.00000 -2.835244e+00       0.000000        0.000000
25%       0.375000     1.00000  3.156273e-01       0.000000        0.000000
50%       0.430556     1.00000  5.406895e-01       0.000000        0.000000
75%       0.500000     1.00000  5.406895e-01       0.000000        1.000000
max       1.000000     2.00000  6.157103e-01       1.000000        1.000000

         treatment  work_interfere   no_employees  remote_work  tech_company  \
count  1250.000000    1.250000e+03  1.250000e+03  1250.000000   1250.000000
mean      0.504800    5.115908e-17 -1.705303e-17     0.298400      0.820000
std       0.500177    1.000400e+00  1.000400e+00     0.457739      0.384341
min       0.000000   -1.826077e+00 -1.603187e+00     0.000000      0.000000
25%       0.000000   -9.679583e-01 -1.027826e+00     0.000000      1.000000
50%       1.000000    7.482798e-01  1.228972e-01     0.000000      1.000000
75%       1.000000    7.482798e-01  6.982587e-01     1.000000      1.000000
max       1.000000    7.482798e-01  1.273620e+00     1.000000      1.000000

       ...    anonymity         leave  mental_health_consequence  \
count  ...  1250.000000  1.250000e+03                1250.000000
```

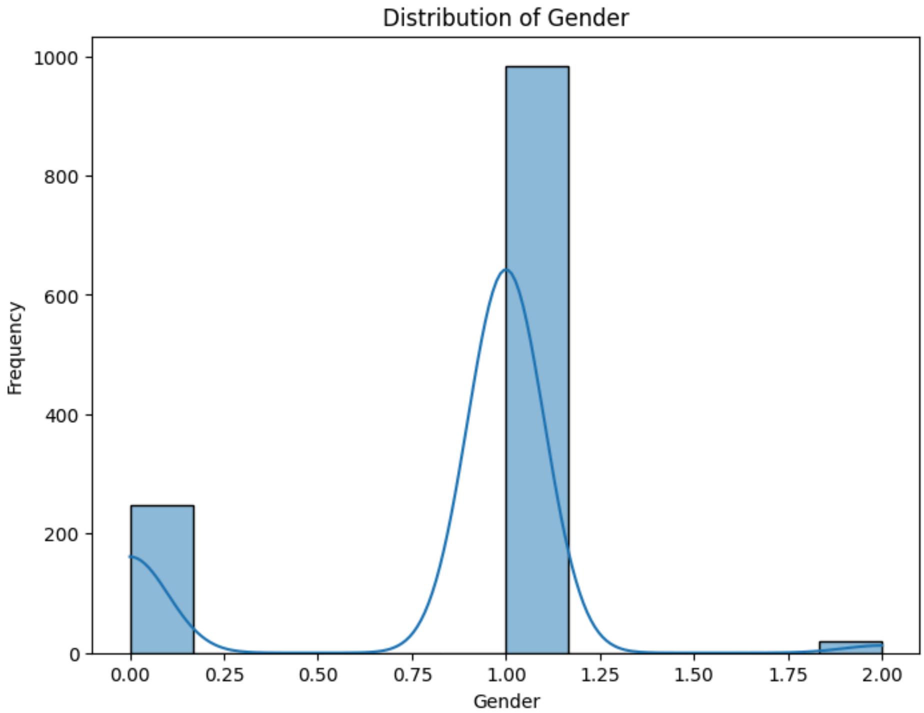
```
mean   ...     0.648000 -8.526513e-18                   0.849600
std    ...     0.909482  1.000400e+00                   0.766453
min    ...     0.000000 -9.346401e-01                   0.000000
25%    ...     0.000000 -9.346401e-01                   0.000000
50%    ...     0.000000 -2.719628e-01                   1.000000
75%    ...     2.000000  3.907145e-01                   1.000000
max    ...     2.000000  1.716069e+00                   2.000000

       phys_health_consequence    coworkers   supervisor  \
count             1250.000000  1250.000000  1250.000000
mean                 0.830400     0.973600     1.100800
std                  0.485205     0.620009     0.843806
min                  0.000000     0.000000     0.000000
25%                  1.000000     1.000000     0.000000
50%                  1.000000     1.000000     1.000000
75%                  1.000000     1.000000     2.000000
max                  2.000000     2.000000     2.000000

       mental_health_interview  phys_health_interview  mental_vs_physical  \
count             1250.000000            1250.000000         1250.000000
mean                 0.868800               0.716000            0.814400
std                  0.425831               0.723715            0.835051
min                  0.000000               0.000000            0.000000
25%                  1.000000               0.000000            0.000000
50%                  1.000000               1.000000            1.000000
75%                  1.000000               1.000000            2.000000
max                  2.000000               2.000000            2.000000

       obs_consequence
count       1250.00000
mean           0.14480
std            0.35204
min            0.00000
25%            0.00000
50%            0.00000
```

Insights:

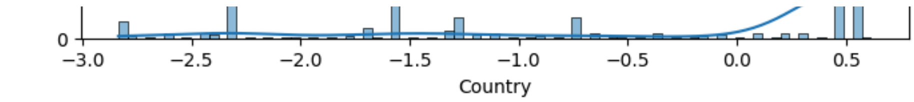
1. **Age:**
   - The mean age is around 44.48% of the maximum age, with a low standard deviation (10.26%), suggesting a relatively narrow age range. The distribution appears positively skewed.
2. **Gender:**
   - Most respondents are coded as Male (mean ≈ 0.82), with a wide standard deviation (42.39%), indicating variability in gender representation. Other gender categories are present.
3. **Country:**
   - The mean value is close to zero, suggesting standardization or normalization. The standard deviation (≈ 1) indicates variability in represented countries.
4. **Other Variables (e.g., treatment, family_history, work_interfere):**
   - These variables exhibit similar patterns of standardization or normalization, with mean values near zero and standard deviations close to one.

```
# Visualize the distribution of numerical variables
for column in numerical_variables:
    plt.figure(figsize=(8, 6))
    sns.histplot(data[column], kde=True)
    plt.title(f'Distribution of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')
    plt.show()
```

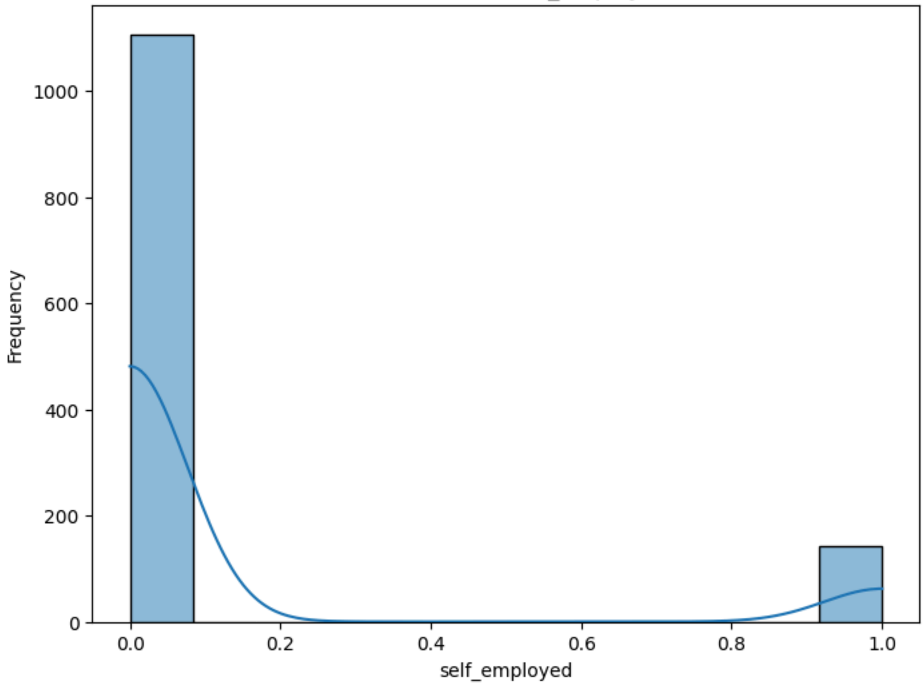

Distribution of Age
Frequency
0
20
40
60
80
100
120
140
0.2
0.4
0.6
0.8
1.0
Age


Distribution of Gender
Frequency
0
200
400
600
800
1000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Gender

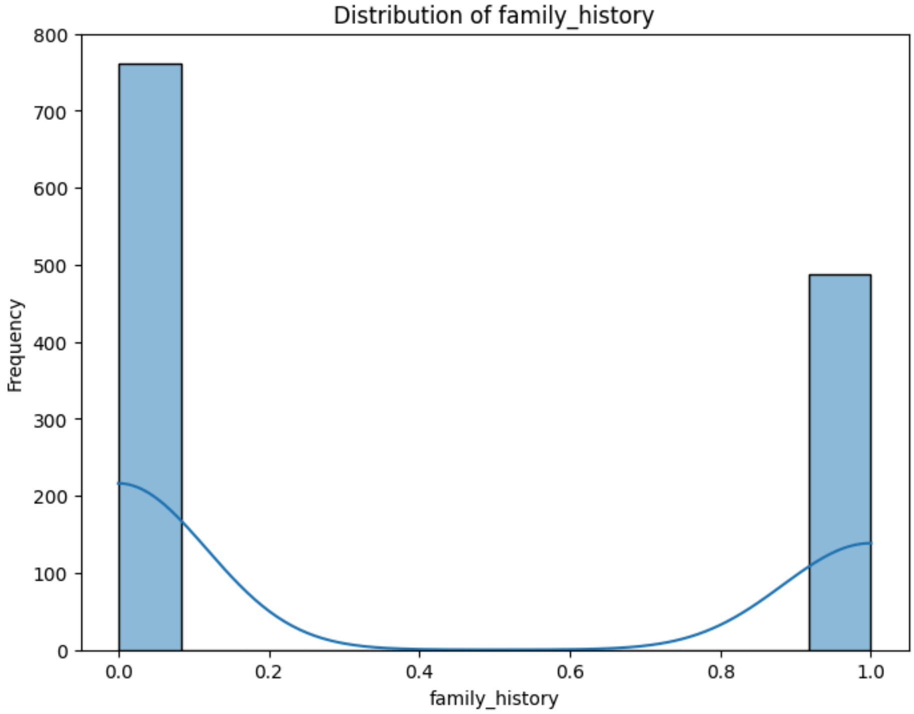

Distribution of Country
Frequency
100
200
300
400
500
600
700



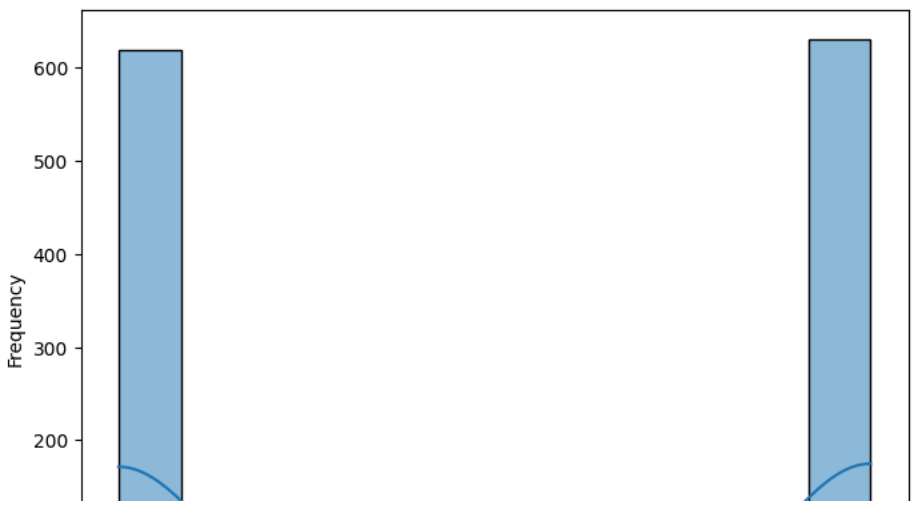
Distribution of self_employed

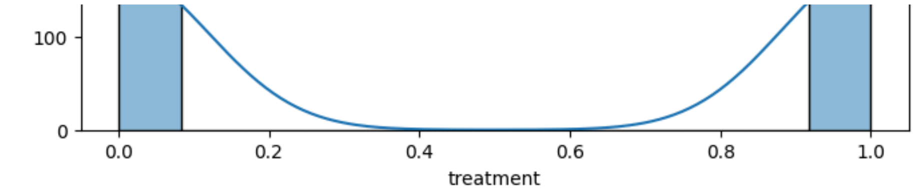


Distribution of family_history

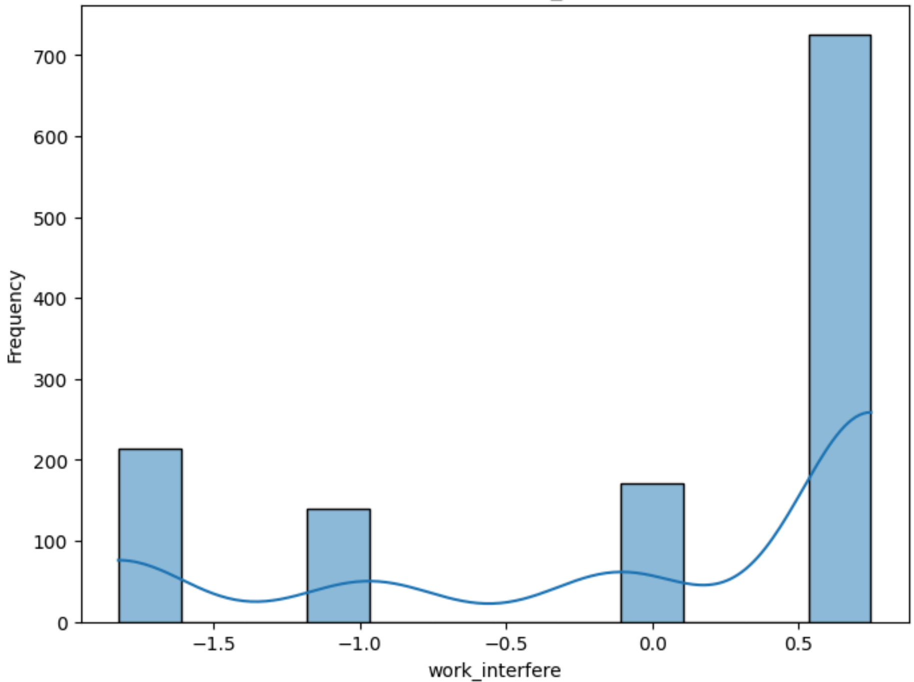


Distribution of treatment

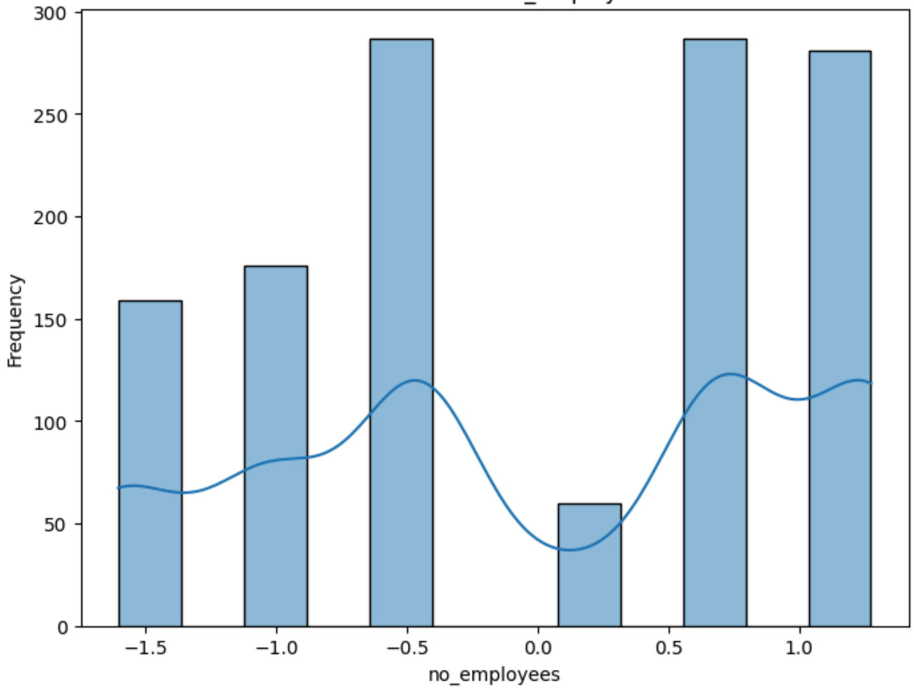




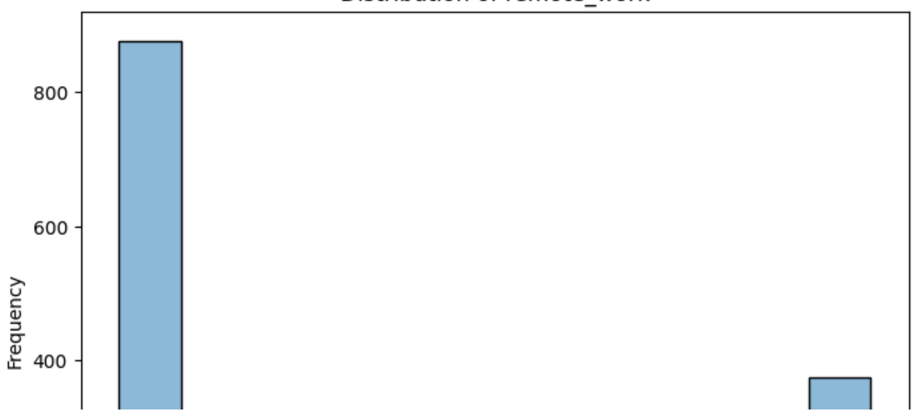
Distribution of work_interfere

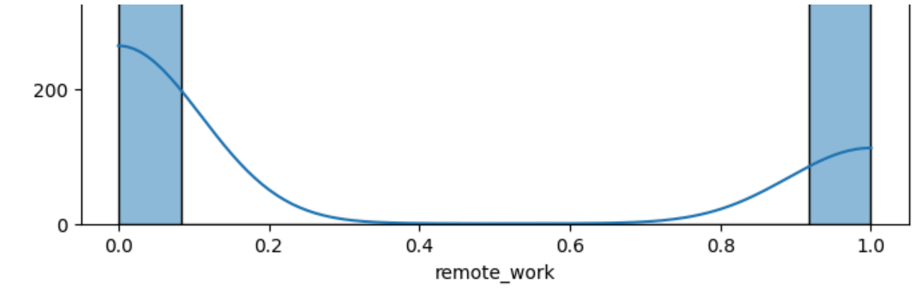


Distribution of no_employees

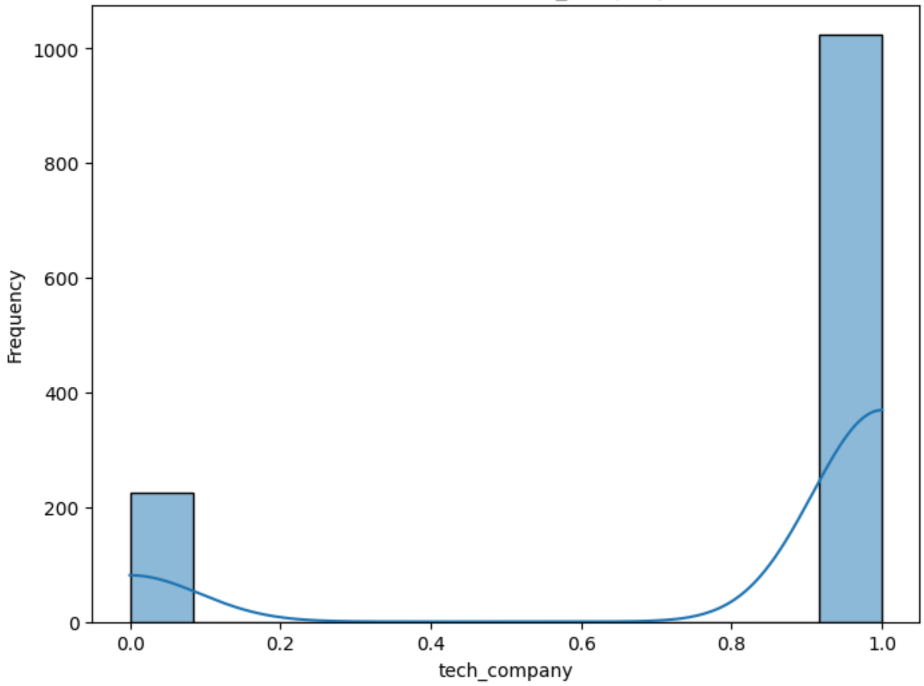


Distribution of remote_work

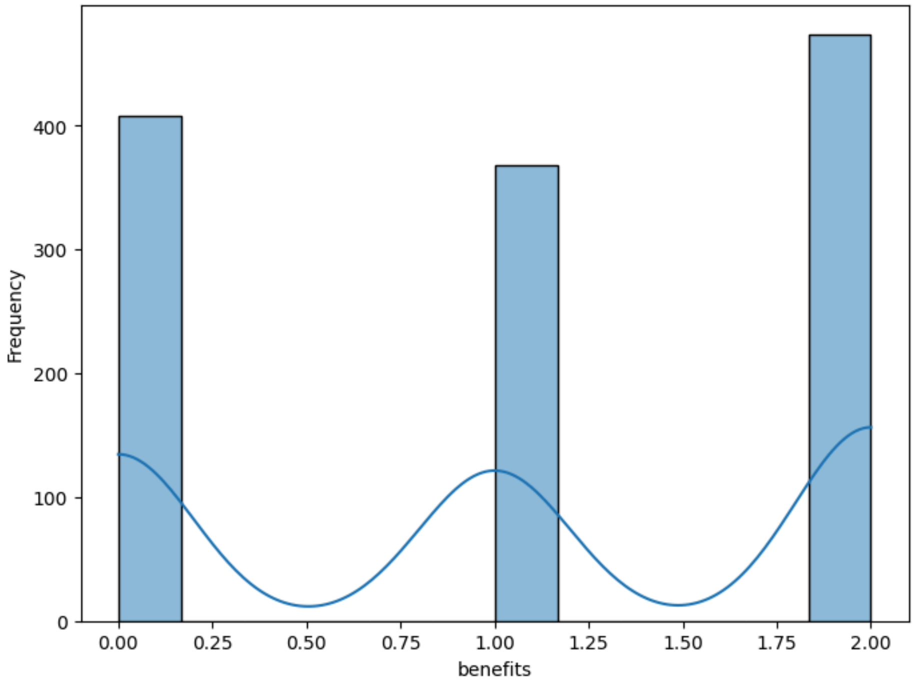




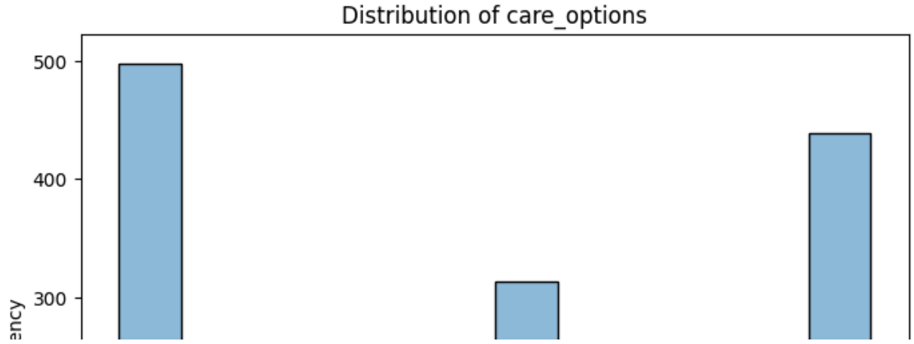
Distribution of tech_company

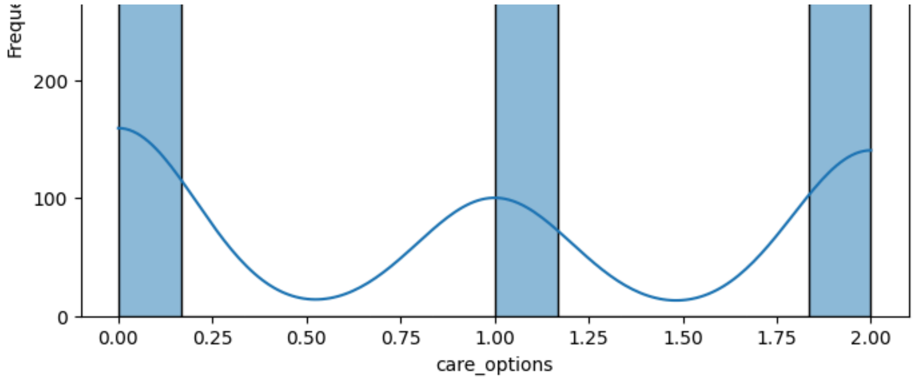


Distribution of benefits

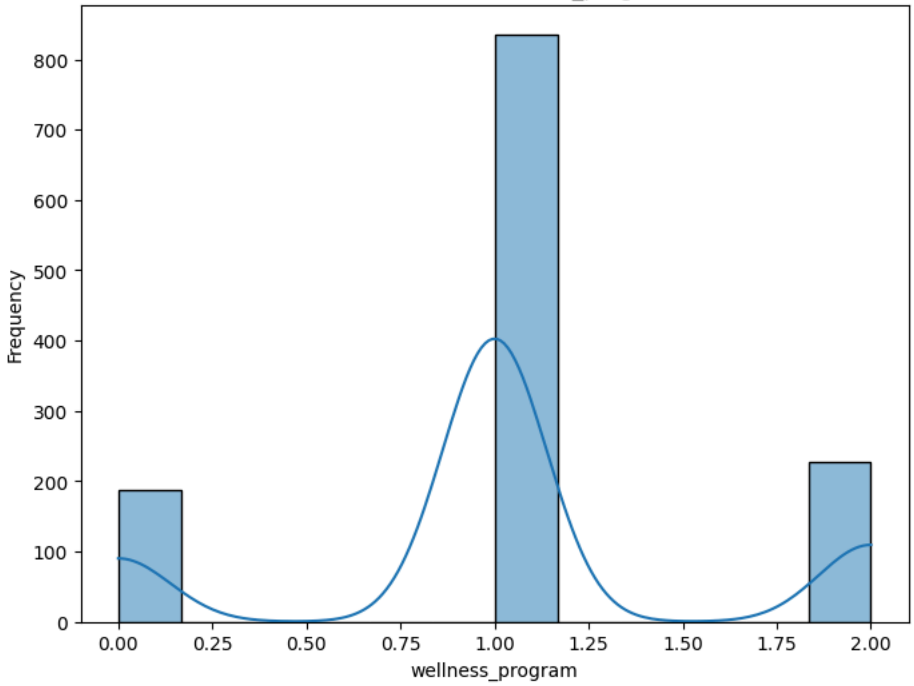


Distribution of care_options

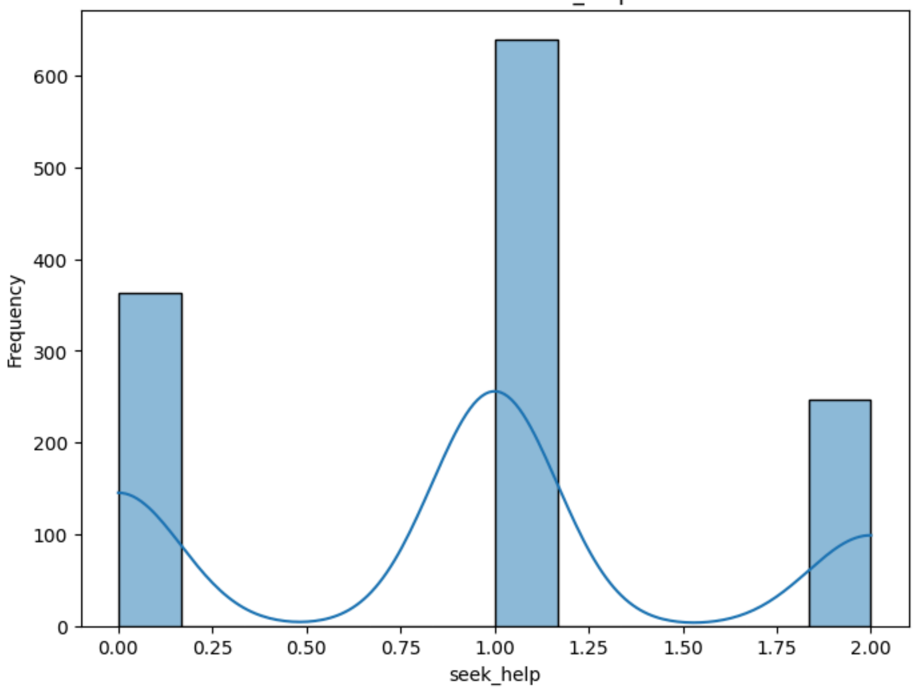




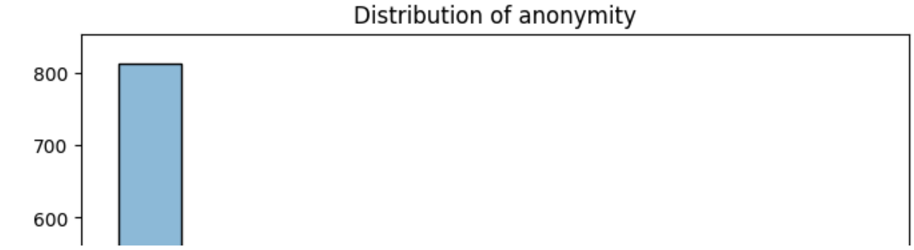
Distribution of wellness_program

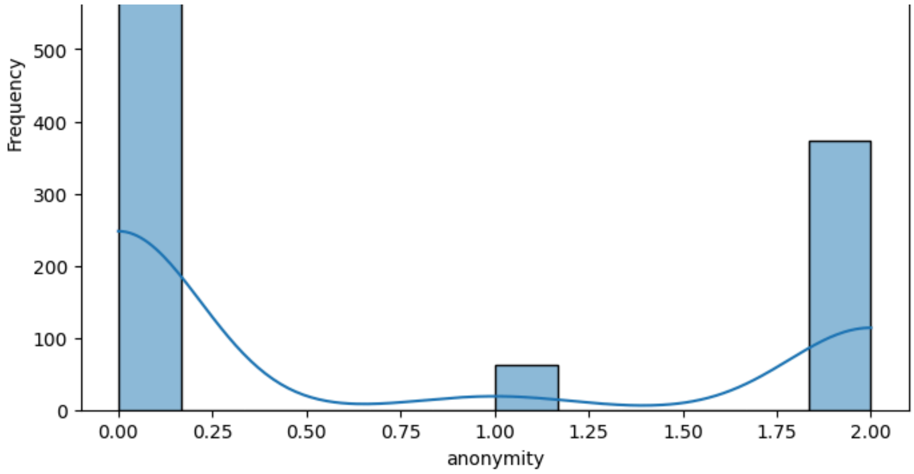


Distribution of seek_help

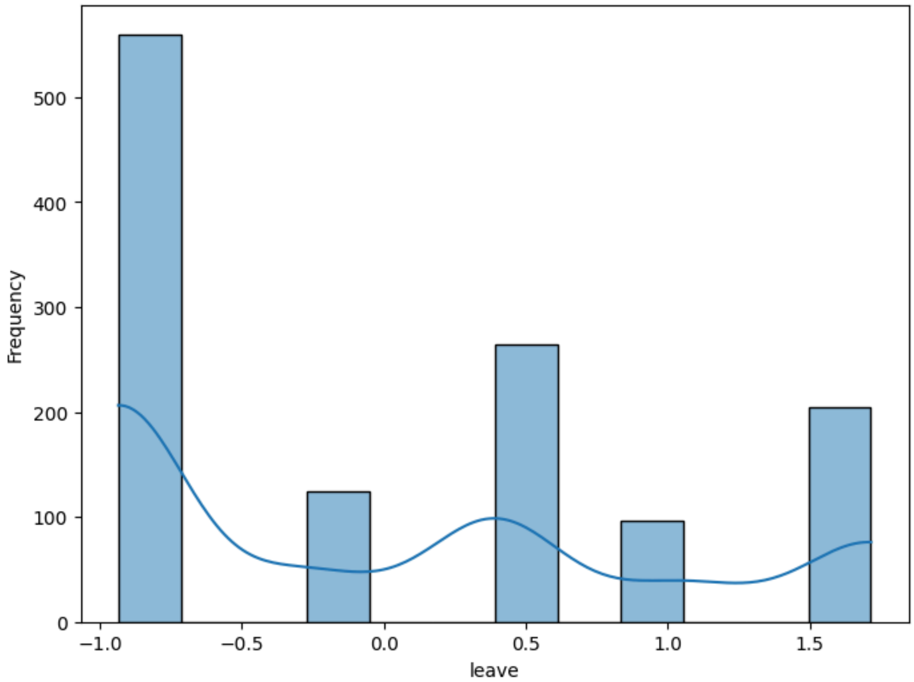


Distribution of anonymity

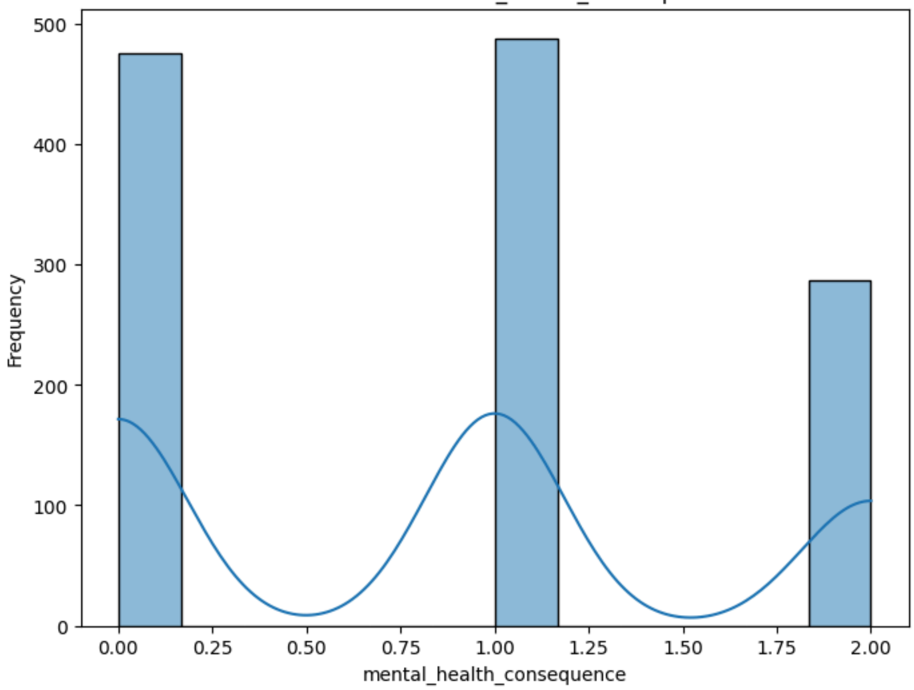




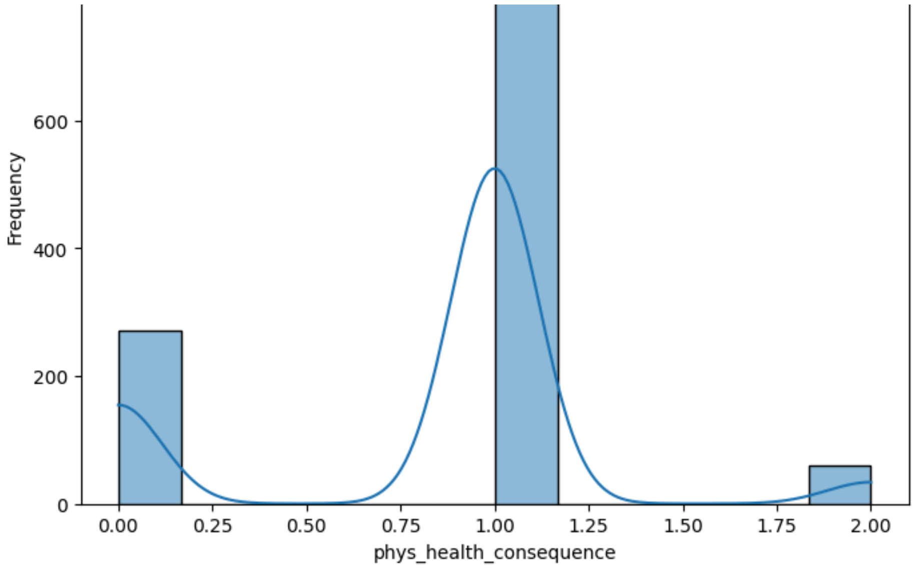
Distribution of leave



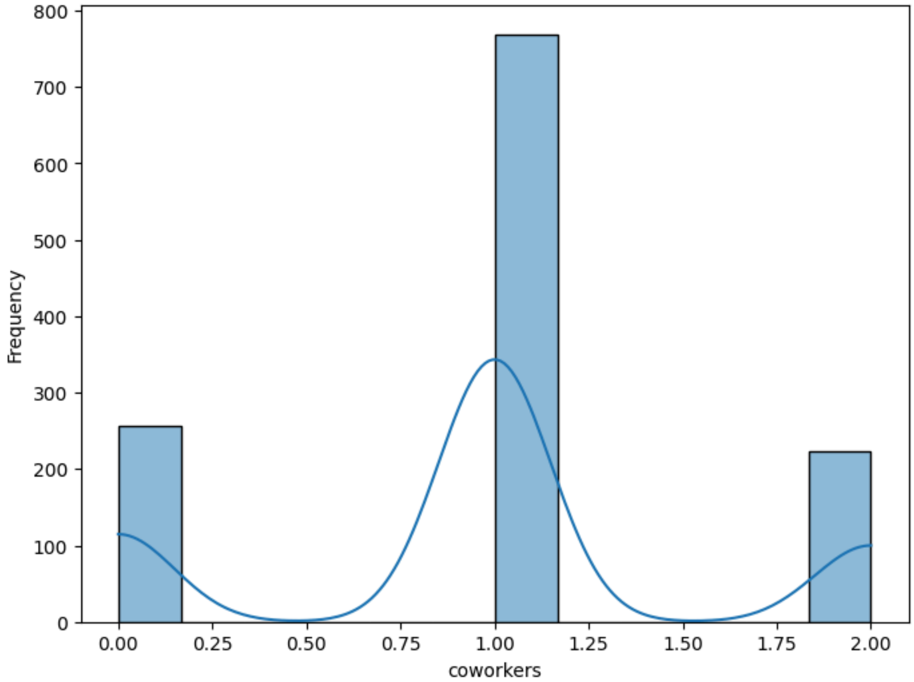
Distribution of mental_health_consequence



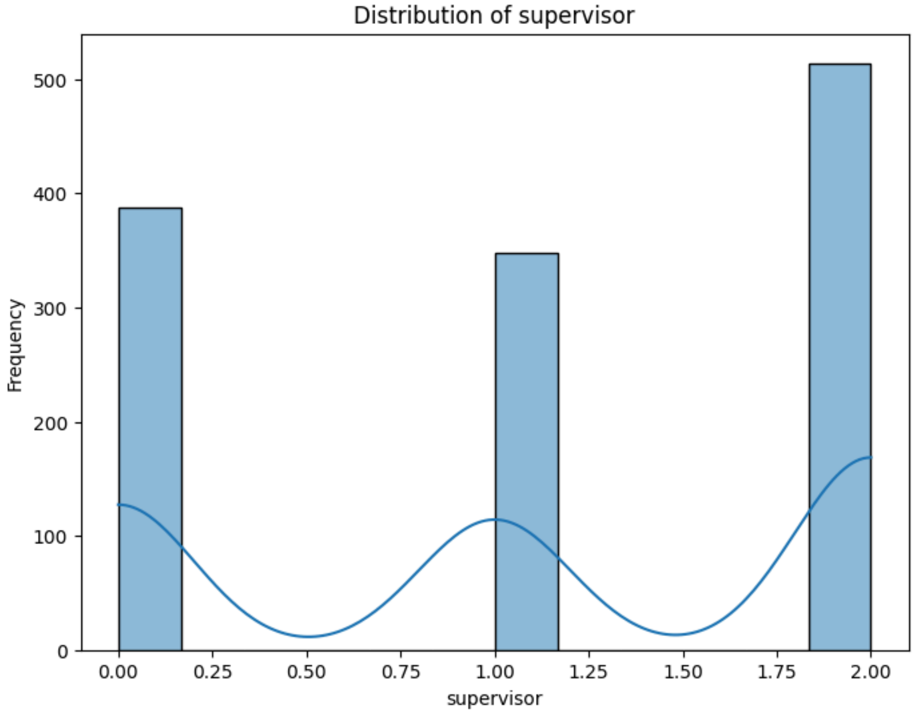
Distribution of phys_health_consequence



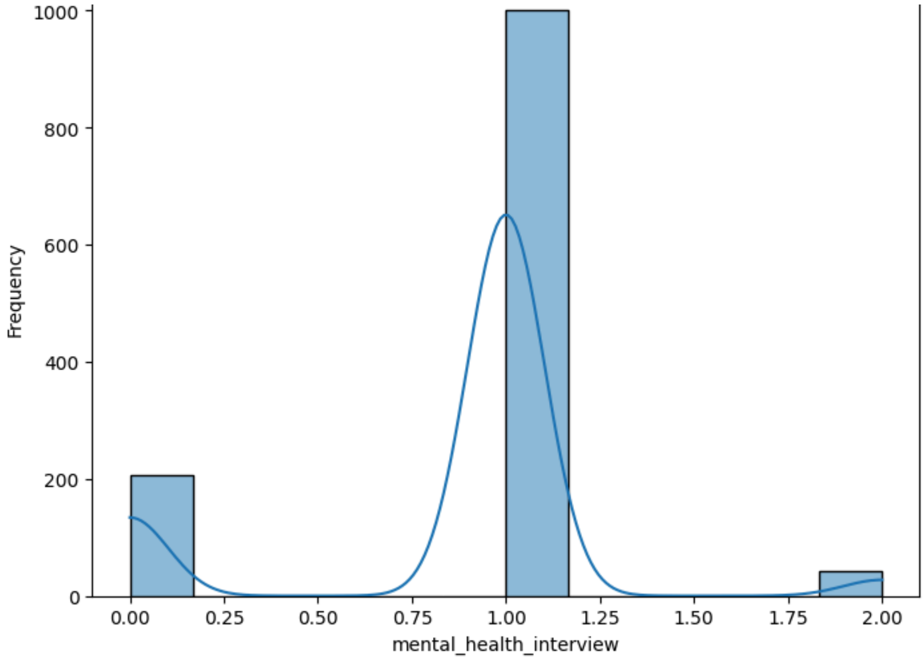
Distribution of coworkers



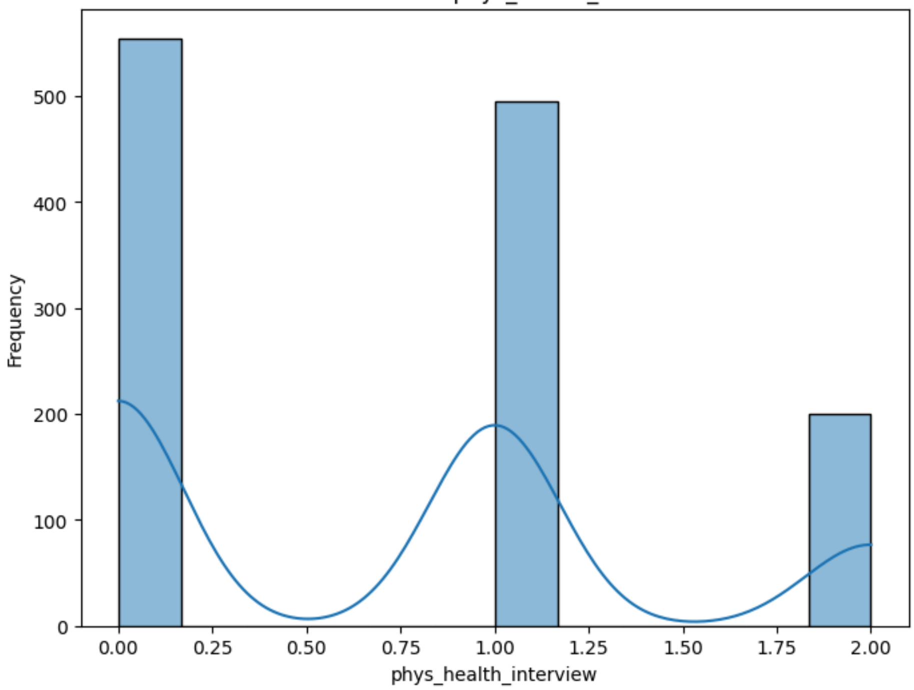
Distribution of supervisor



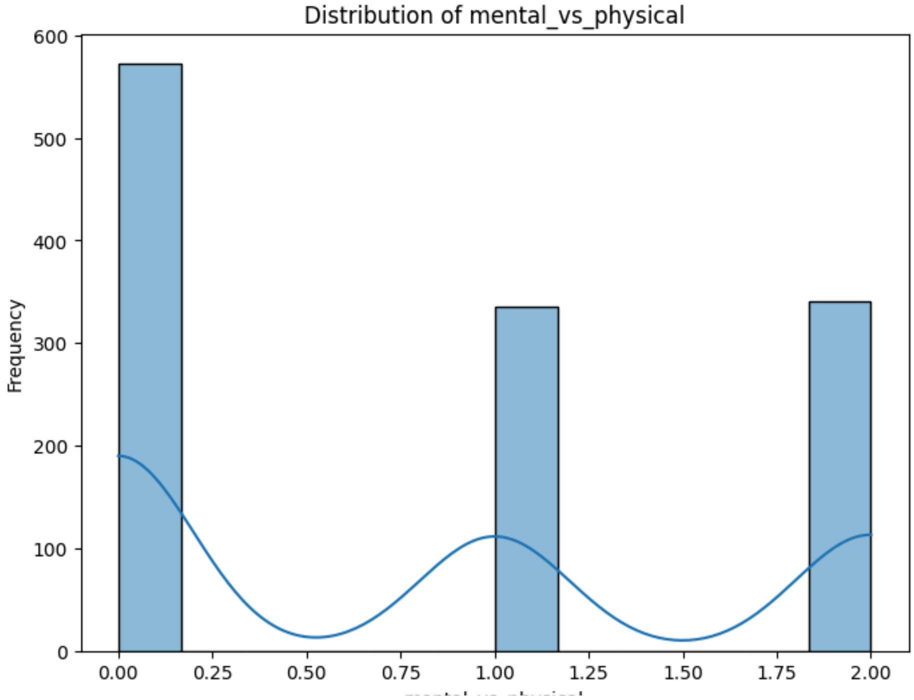
Distribution of mental_health_interview



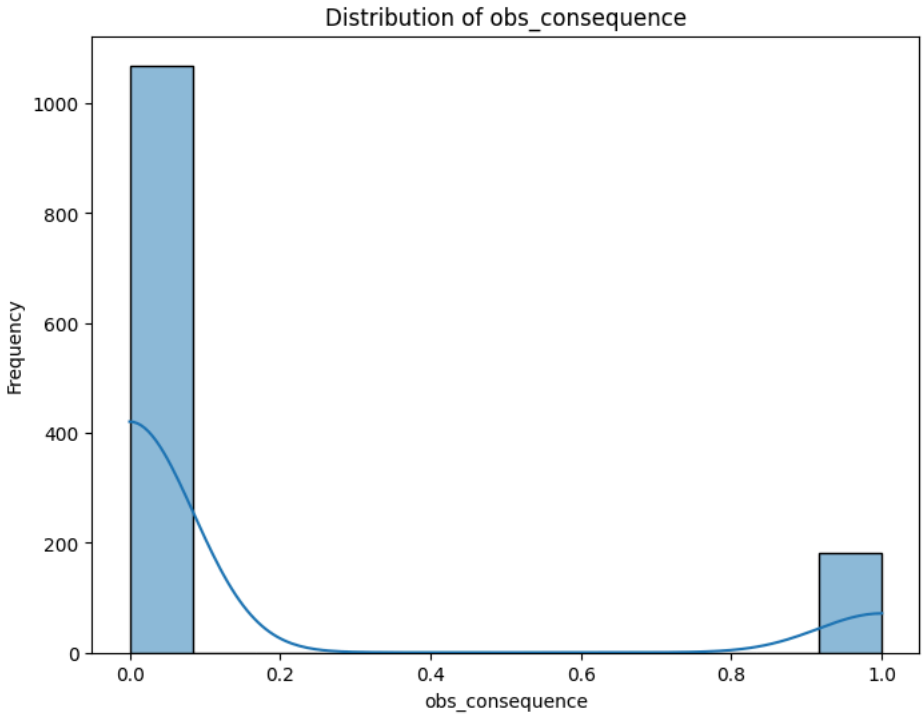
Distribution of phys_health_interview



Distribution of mental_vs_physical



mental_vs_physical



Insights:

1.**Age:**

- The age distribution appears to be slightly positively skewed, with a peak around the middle of the range. The KDE plot suggests a relatively smooth distribution, indicating that there are no sharp spikes or outliers. Certainly! Let's delve deeper into the inferences for the other numerical variables:

  1. **Gender:**
     - The distribution of gender, while numerical after label encoding, should ideally represent different categories. However, the mean value being around 0.82 indicates that the majority of the respondents are encoded as 1, which might correspond to a specific gender category. This warrants further investigation to ensure proper encoding and representation of gender categories.
  2. **Self-employed:**
     - The mean value of approximately 0.11 suggests that a small proportion of respondents identify as self-employed based on the label encoding. This indicates that the majority of respondents in the dataset are not self-employed.
  3. **Family History:**
     - The mean value of around 0.39 indicates that a significant portion of respondents have a family history of mental illness, as encoded in the dataset. This variable's distribution is binary, indicating the presence or absence of a family history of mental illness.
  4. **Treatment:**
     - The mean value of approximately 0.50 suggests that the dataset is balanced in terms of respondents who have sought treatment for mental health conditions. This balanced distribution is crucial for training classification models without bias towards any particular class.
  5. **Work Interference:**
     - The mean value being close to 0 indicates that the distribution of work interference with mental health conditions might be evenly spread across the dataset after preprocessing. This variable's distribution likely represents different levels of interference with work due to mental health issues.
  6. **Other Variables:**
     - Similarly, for other numerical variables such as no_employees, remote_work, and tech_company, the mean values provide insights into their distributions after preprocessing. These variables may represent different aspects of respondents' work environments or organizational characteristics.

Overall, analyzing the numerical variables' distributions and summary statistics helps understand their characteristics and the impact of preprocessing steps on the dataset. These inferences aid in further exploratory data analysis and model building processes.