D606 Task 2 – Data Science Capstone – Data Analysis Report

Student: Nichole Gonzales

Table of Contents

Project Name: Binary Logistic Regression on Employee Burnout Dataset

Section A: Research Question

The purpose of this study is to evaluate whether a binary logistic regression model can be used to classify employees as being a high or low risk of Burnout using workplace data. The research question is: *Can a logistic regression model classify employees as high or low risk of Burnout using workplace data?*

The context for this research is grounded in the prevalence of employee burnout in modern workplaces. Employee burnout is a concern in modern-day workplaces; half (52%) of employees reported feeling symptoms of Burnout (NAMI, 2024). Forbes reported that two-thirds (66%) of American employees experienced Burnout in 2025 (Robinson, 2025). Prior research by Liu et al. collected responses using a five-point Likert scale survey asking employees to measure their own job satisfaction. The results of the survey were analyzed by Liu et al. to identify employees exhibiting signs of Burnout accurately (with a close to 80% accuracy). This supports the feasibility of classification-based approaches in organizational settings (Liu et al., 2025).

Based on this context, this study seeks to demonstrate how statistical modeling can be applied to workplace data to support early identification of burnout risk. The findings of this analysis may inform workplace strategies aimed at improving employee well-being and retention.

The null hypothesis states that a logistic regression model cannot predict whether employees are at high or low risk of Burnout using the collected workplace data. The alternative hypothesis states that a logistic regression model can predict whether employees are at a high or low risk of Burnout with an accuracy greater than 70%.

Section B: Data Collection

The data used in this study was obtained from Harvard Dataverse and is titled "Burnout Among Corporate Employees" (Devkar, 2025). The dataset contains survey responses that provide de-identified, generalized information about the workplace, behavioral insights, and employee well-being. The dataset contains 22,750 observations and 17 variables prior to data cleaning and preparation. Variables include both continuous and categorical data. The use of structured data enables quantitative analysis and supports the application of statistical classification techniques.

An advantage of using a publicly available dataset is that the analysis could be independently reproduced, supporting transparency and potential peer review. Additionally, the dataset was sourced from a reputable scholarly data repository, which provides confidence in the structural integrity and consistency of the data, making it suitable for study.

A disadvantage of this data-collection methodology is that the data are self-reported, which introduces the potential for response bias. Employees may underreport or overreport aspects of their well-being, workload, or work conditions due to perception, recall limitations, or social desirability.

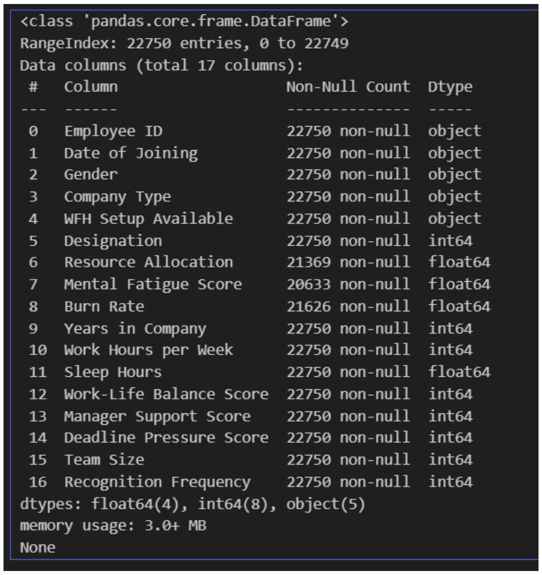
A challenge encountered during data collection was the absence of a formal data dictionary or documentation detailing the specific survey questions used to generate each variable. To address this challenge, exploratory data analysis was conducted following data extraction. Variable distributions, value ranges, and relationships were examined, along with domain knowledge related to workplace surveys, which was applied to infer the likely meaning and intent of each variable. This approach enabled informed interpretation of the data while maintaining analytical consistency throughout the study.

Section C. Data Extraction and Preparation

The dataset used in this study was extracted from the Harvard Database in comma-separated values (CSV) format. Python was used as the primary tool for extraction and preparation within a Jupyter Notebook environment in Visual Studio Code. The CSV file was imported into Python using the Pandas library, which enabled efficient data manipulation, exploration, and transformation. An advantage of using Python and Pandas for data extraction and preparation is the ability to perform reproducible, transparent, and scalable data transformations. The Jupyter Notebook environment enables the documentation and verification of each step in the workflow, supporting analytical rigor and traceability. A disadvantage of this approach is that data preparation decisions, such as imputation and variable encoding, require justification to prevent the introduction of unintended bias.

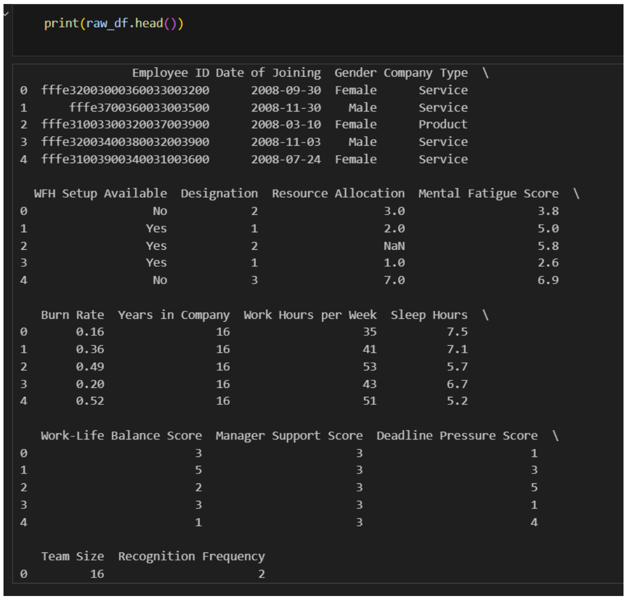
Following the extraction, an initial exploratory assessment of the dataset was conducted to evaluate its structure, data types, and completeness. Descriptive statistics and dimensional checks were performed to confirm the number of records and variables prior to cleaning. Visualization of missing data patterns was conducted using the Missingno library, which provided insight into the extent and any possible patterns in the missing values across variables. Understanding Missingness in the dataset supports informed decision-making for data treatment strategies. The following screenshots illustrate the steps for initial data exploration.

Using Pandas info to understand the structure, data types, and completeness:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22750 entries, 0 to 22749
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Employee ID              22750 non-null  object 
 1   Date of Joining          22750 non-null  object 
 2   Gender                   22750 non-null  object 
 3   Company Type             22750 non-null  object 
 4   WFH Setup Available      22750 non-null  object 
 5   Designation              22750 non-null  int64  
 6   Resource Allocation      21369 non-null  float64
 7   Mental Fatigue Score     20633 non-null  float64
 8   Burn Rate                21626 non-null  float64
 9   Years in Company         22750 non-null  int64  
 10  Work Hours per Week      22750 non-null  int64  
 11  Sleep Hours              22750 non-null  float64
 12  Work-Life Balance Score  22750 non-null  int64  
 13  Manager Support Score    22750 non-null  int64  
 14  Deadline Pressure Score  22750 non-null  int64  
 15  Team Size                22750 non-null  int64  
 16  Recognition Frequency    22750 non-null  int64  
dtypes: float64(4), int64(8), object(5)
memory usage: 3.0+ MB
None
```

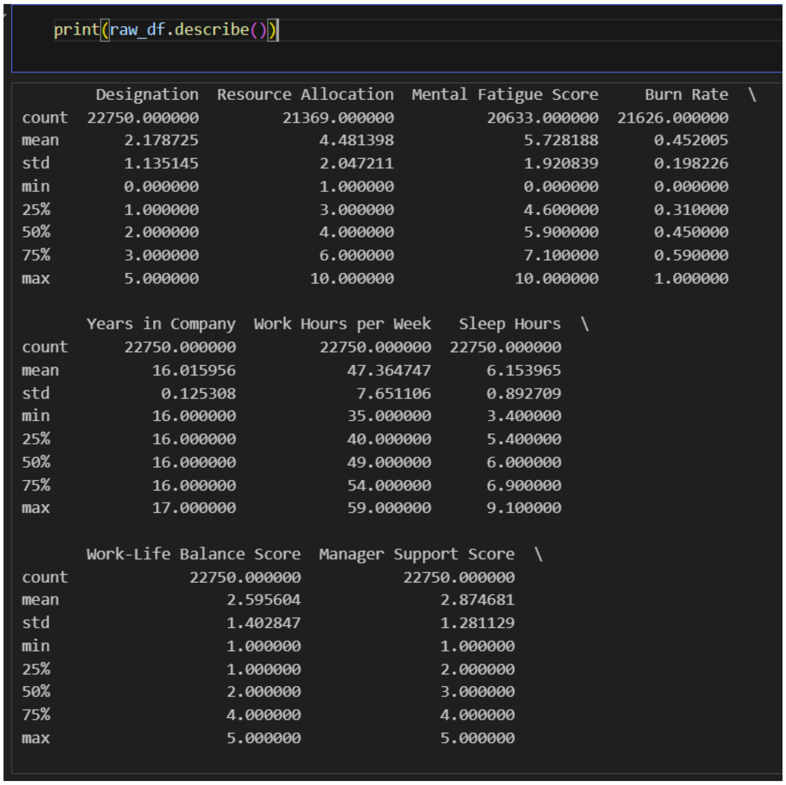
Using Pandas Head to view examples of records:



```
print(raw_df.head())
```

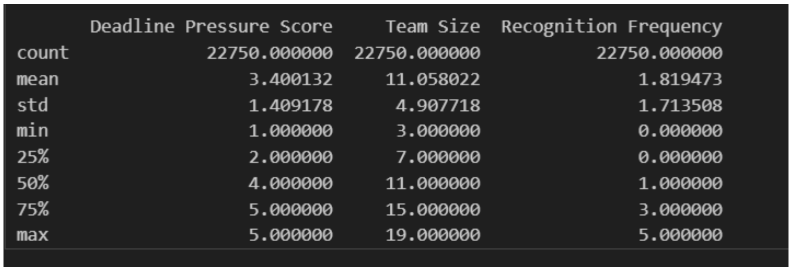
```
                Employee ID Date of Joining  Gender Company Type  \
0  fffe32003000360033003200      2008-09-30  Female      Service   
1      fffe3700360033003500      2008-11-30    Male      Service   
2  fffe31003300320037003900      2008-03-10  Female      Product   
3  fffe32003400380032003900      2008-11-03    Male      Service   
4  fffe31003900340031003600      2008-07-24  Female      Service   

  WFH Setup Available  Designation  Resource Allocation  Mental Fatigue Score  \
0                  No            2                  3.0                   3.8   
1                 Yes            1                  2.0                   5.0   
2                 Yes            2                  NaN                   5.8   
3                 Yes            1                  1.0                   2.6   
4                  No            3                  7.0                   6.9   

   Burn Rate  Years in Company  Work Hours per Week  Sleep Hours  \
0       0.16                16                   35          7.5   
1       0.36                16                   41          7.1   
2       0.49                16                   53          5.7   
3       0.20                16                   43          6.7   
4       0.52                16                   51          5.2   

   Work-Life Balance Score  Manager Support Score  Deadline Pressure Score  \
0                        3                      3                        1   
1                        5                      3                        3   
2                        2                      3                        5   
3                        3                      3                        1   
4                        1                      3                        4   

   Team Size  Recognition Frequency  
0         16                      2  
```

View descriptive statistics of numeric variables with Pandas describe:



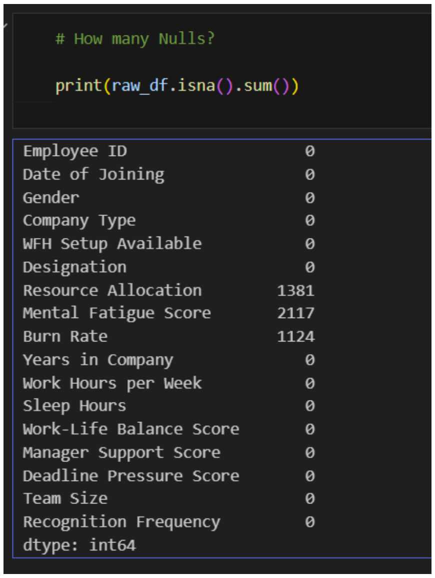
```
print(raw_df.describe())
```

```
       Designation  Resource Allocation  Mental Fatigue Score     Burn Rate  \
count 22750.000000         21369.000000          20633.000000  21626.000000
mean      2.178725             4.481398              5.728188      0.452005
std       1.135145             2.047211              1.920839      0.198226
min       0.000000             1.000000              0.000000      0.000000
25%       1.000000             3.000000              4.600000      0.310000
50%       2.000000             4.000000              5.900000      0.450000
75%       3.000000             6.000000              7.100000      0.590000
max       5.000000            10.000000             10.000000      1.000000

       Years in Company  Work Hours per Week  Sleep Hours  \
count      22750.000000         22750.000000 22750.000000
mean          16.015956            47.364747     6.153965
std            0.125308             7.651106     0.892709
min           16.000000            35.000000     3.400000
25%           16.000000            40.000000     5.400000
50%           16.000000            49.000000     6.000000
75%           16.000000            54.000000     6.900000
max           17.000000            59.000000     9.100000

       Work-Life Balance Score  Manager Support Score  \
count             22750.000000           22750.000000
mean                  2.595604               2.874681
std                   1.402847               1.281129
min                   1.000000               1.000000
25%                   1.000000               2.000000
50%                   2.000000               3.000000
75%                   4.000000               4.000000
max                   5.000000               5.000000
```



```
       Deadline Pressure Score     Team Size  Recognition Frequency
count             22750.000000  22750.000000           22750.000000
mean                  3.400132     11.058022               1.819473
std                   1.409178      4.907718               1.713508
min                   1.000000      3.000000               0.000000
25%                   2.000000      7.000000               0.000000
50%                   4.000000     11.000000               1.000000
75%                   5.000000     15.000000               3.000000
max                   5.000000     19.000000               5.000000
```

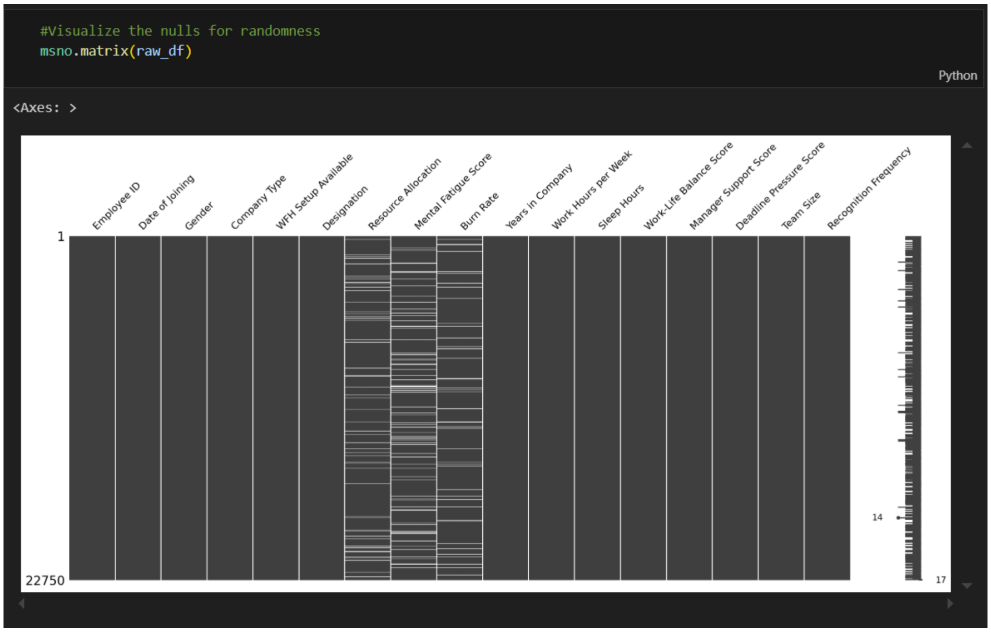
Obtain a count of how many nulls are present in the dataset for each variable:



```
# How many Nulls?

print(raw_df.isna().sum())
```

```
Employee ID                   0
Date of Joining               0
Gender                        0
Company Type                  0
WFH Setup Available           0
Designation                   0
Resource Allocation        1381
Mental Fatigue Score       2117
Burn Rate                  1124
Years in Company              0
Work Hours per Week           0
Sleep Hours                   0
Work-Life Balance Score       0
Manager Support Score         0
Deadline Pressure Score       0
Team Size                     0
Recognition Frequency         0
dtype: int64
```

Visualize Missingness with the missingno matrix:



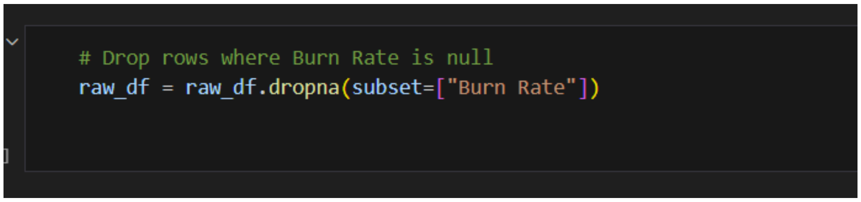
```
#Visualize the nulls for randomness
msno.matrix(raw_df)
```

```
<Axes: >
```

Following data exploration, the dataset needed to be prepared for binary logistic regression. There were multiple steps to this process to ensure the data were suitable for the study's goal, including treating missing variables and transforming other variables.

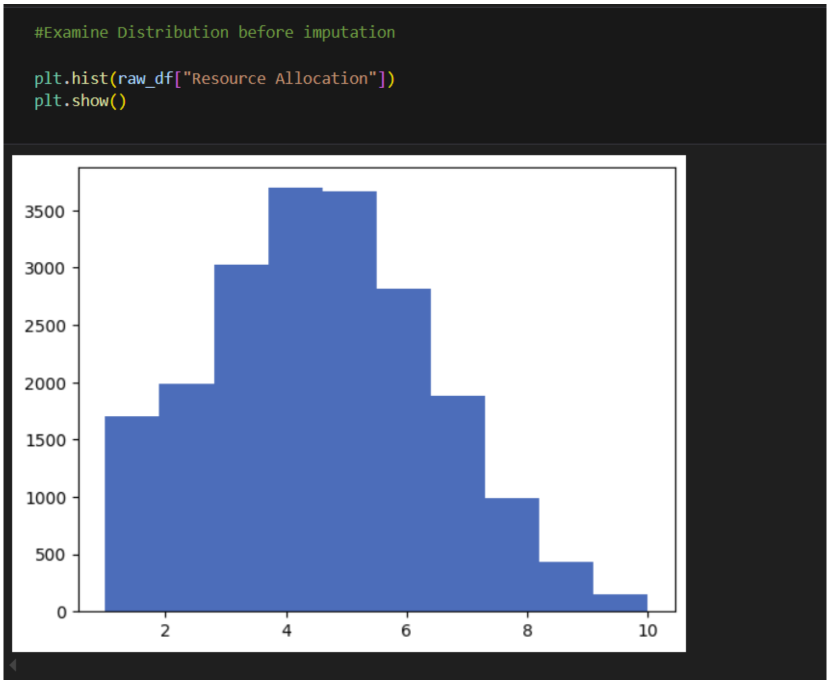
The missing variables Resource Allocation, Mental Fatigue Score, and Burn Rate required treatment. The records with missing values for the "Burn Rate" variable were dropped from the dataset prior to analysis, the advantage being to preserve the integrity of the target variable. Dropping the missing records for "Burn Rate" was not detrimental to this study, as 95% of the total dataset shall be preserved. A disadvantage to dropping these records is that it reduces the total number of records. The “Burn Rate” will be used to create the target variable for predicting burnout. The creation of the target variable will be discussed with other transformative steps. Missing values for the “Resource Allocation” and “Mental Fatigue Score” variables were addressed using median imputation. The distribution of these variables was examined prior to imputation. An advantage of median imputation is its robustness to skewed distributions and outliers, allowing the central tendency of each variable to be preserved (Samal, 2024). A disadvantage of median imputation is that it reduces natural variability within the data, potentially obscuring subtle relationships between predictors and the target variable. The distribution was re-examined after imputation to measure any shift in skewness. The following screenshots support the steps mentioned for data preparation and treatment of null variables.

Dropping the rows where “Burn Rate” is null:



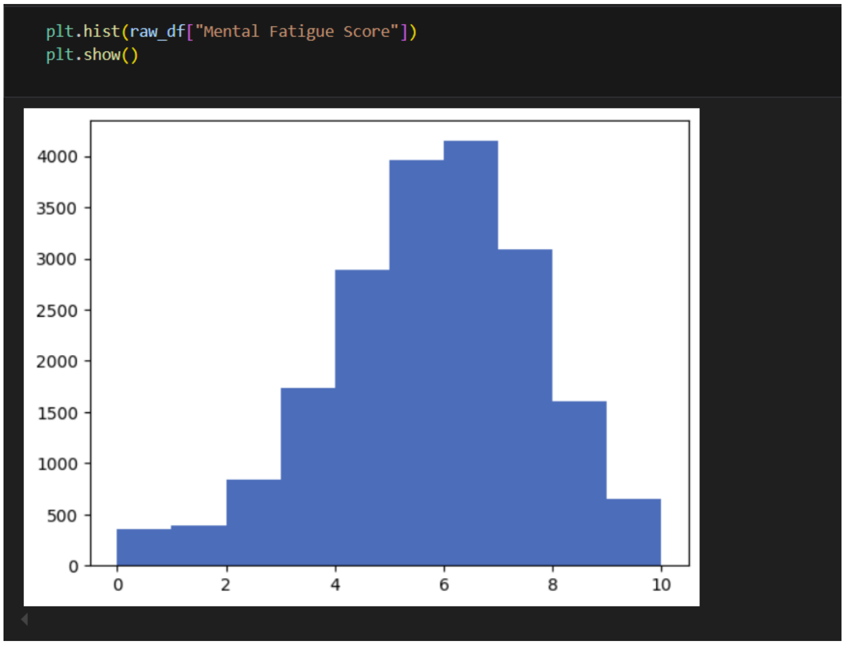
```
# Drop rows where Burn Rate is null
raw_df = raw_df.dropna(subset=["Burn Rate"])
```

Histogram of “Resource Allocation” prior to imputation:



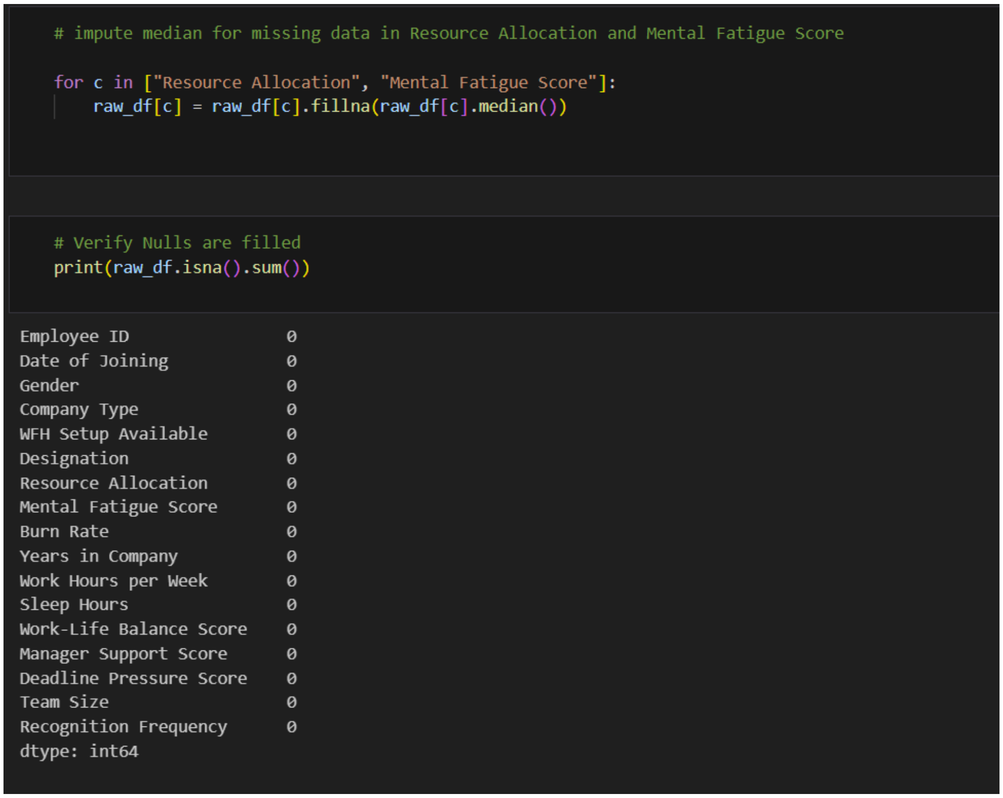
```
#Examine Distribution before imputation

plt.hist(raw_df["Resource Allocation"])
plt.show()
```

Histogram of “Mental Fatigue Score” prior to imputation:



```
plt.hist(raw_df["Mental Fatigue Score"])
plt.show()
```

Code to impute median for missing data in “Resource Allocation”, “Mental Fatigue Score”, and verify the nulls are filled:

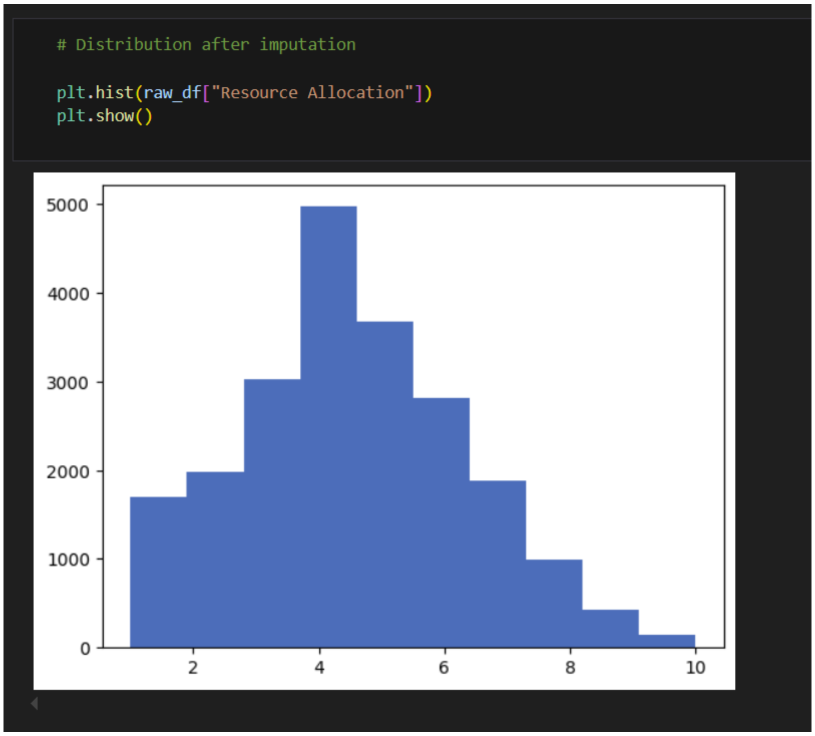


```
# impute median for missing data in Resource Allocation and Mental Fatigue Score

for c in ["Resource Allocation", "Mental Fatigue Score"]:
    raw_df[c] = raw_df[c].fillna(raw_df[c].median())
```

```
# Verify Nulls are filled
print(raw_df.isna().sum())
```

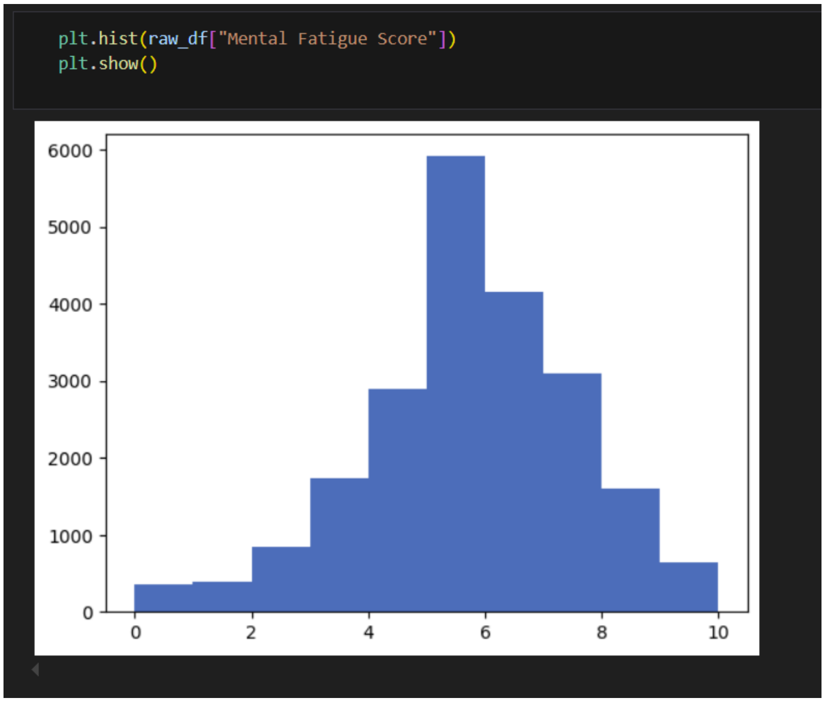
```
Employee ID                0
Date of Joining            0
Gender                     0
Company Type               0
WFH Setup Available        0
Designation                0
Resource Allocation        0
Mental Fatigue Score       0
Burn Rate                  0
Years in Company           0
Work Hours per Week        0
Sleep Hours                0
Work-Life Balance Score    0
Manager Support Score      0
Deadline Pressure Score    0
Team Size                  0
Recognition Frequency      0
dtype: int64
```

Histogram of “Resource Allocation” after imputation:



```python
# Distribution after imputation

plt.hist(raw_df["Resource Allocation"])
plt.show()
```

Histogram of “Mental Fatigue Score” after imputation:

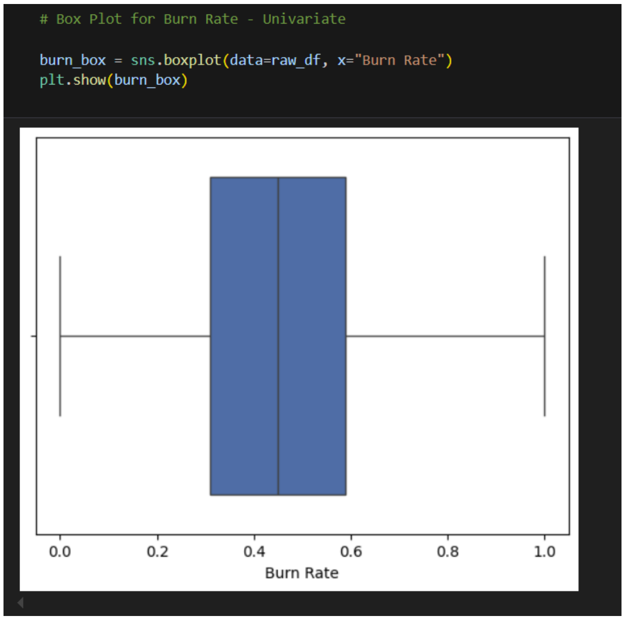


```python
plt.hist(raw_df["Mental Fatigue Score"])
plt.show()
```

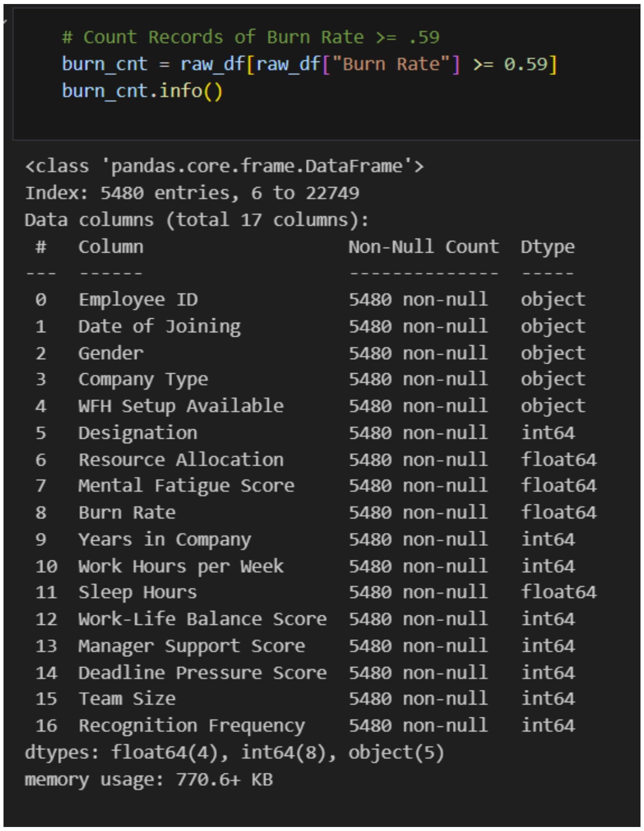
After the missing values are treated, data exploration continues by understanding the distribution of all numeric variables with the use of a univariate graph (box plot).

The first variable examined is the one that will become the target variable in this study, “Burn Rate”. This boxplot provides a visualization of where the 4th quartile range begins, which will serve as the starting point for identifying employees at high risk of Burnout. This statistic was provided earlier in the descriptive statistics results for the 4th quartile, starting at a Burn Rate of 0.59. An advantage of choosing this threshold method to identify high and low risk employees is that it focuses attention on employees with the highest observed levels of Burnout. The disadvantage of using a quartile-based threshold is that it will classify employees on the cusp of Burnout differently, despite having similar underlying risk profiles.

A count was also obtained to understand how many records in the treated population fall in the range of high risk of Burnout.



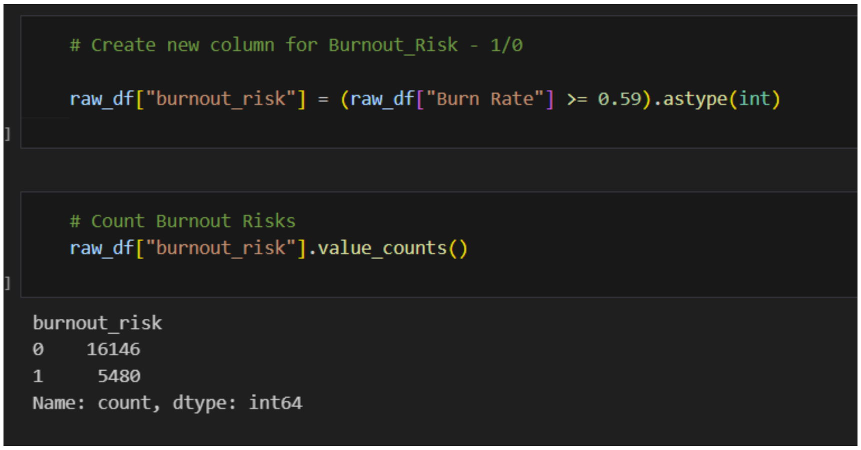
```
# Box Plot for Burn Rate - Univariate

burn_box = sns.boxplot(data=raw_df, x="Burn Rate")
plt.show(burn_box)
```



```
# Count Records of Burn Rate >= .59
burn_cnt = raw_df[raw_df["Burn Rate"] >= 0.59]
burn_cnt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5480 entries, 6 to 22749
Data columns (total 17 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   Employee ID            5480 non-null   object 
 1   Date of Joining        5480 non-null   object 
 2   Gender                 5480 non-null   object 
 3   Company Type           5480 non-null   object 
 4   WFH Setup Available    5480 non-null   object 
 5   Designation            5480 non-null   int64  
 6   Resource Allocation    5480 non-null   float64
 7   Mental Fatigue Score   5480 non-null   float64
 8   Burn Rate              5480 non-null   float64
 9   Years in Company       5480 non-null   int64  
 10  Work Hours per Week    5480 non-null   int64  
 11  Sleep Hours            5480 non-null   float64
 12  Work-Life Balance Score  5480 non-null   int64  
 13  Manager Support Score  5480 non-null   int64  
 14  Deadline Pressure Score  5480 non-null   int64  
 15  Team Size              5480 non-null   int64  
 16  Recognition Frequency  5480 non-null   int64  
dtypes: float64(4), int64(8), object(5)
memory usage: 770.6+ KB
```

Preparing the target datapoint further required a new column to be created, where if the "Burn Rate" was greater than or equal to 0.59, a 1 shall be imputed to signify an employee at high risk of Burnout, otherwise a 0 shall be imputed for a "Burn Rate" less than 0.59 to signify an employee at low risk of Burnout. A count of records was performed to verify that the numbers were properly imputed.



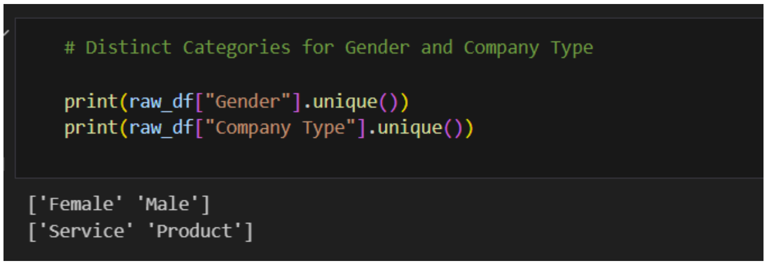
```
# Create new column for Burnout_Risk - 1/0

raw_df["burnout_risk"] = (raw_df["Burn Rate"] >= 0.59).astype(int)
```

```
# Count Burnout Risks
raw_df["burnout_risk"].value_counts()
```

```
burnout_risk
0    16146
1     5480
Name: count, dtype: int64
```

Categorical variables that required transformation were "WFH Setup Available", "Gender", and "Company Type". The values of these variables were examined using Pandas' Unique feature to provide a distinct listing of the types of categories, each of which revealed only two types recorded under them. Based on this outcome, their variables were transformed into binary variables. The binary categorical variable "WFH Setup Available", originally recorded as "Yes" or "No," was recoded into numeric values, where 1 represents "Yes" and 0 represents "No," to support compatibility with machine learning algorithms. "Gender" and "Company Type" are also binary variables. "Gender" has two possibilities: “Male” or “Female”. These will be recoded where 1 represents Male and 0 represents Female. The "Company Type" has two possibilities: "Service" or "Product". These will be recoded, where 1 represents "Product” and 0 represents “Service”.

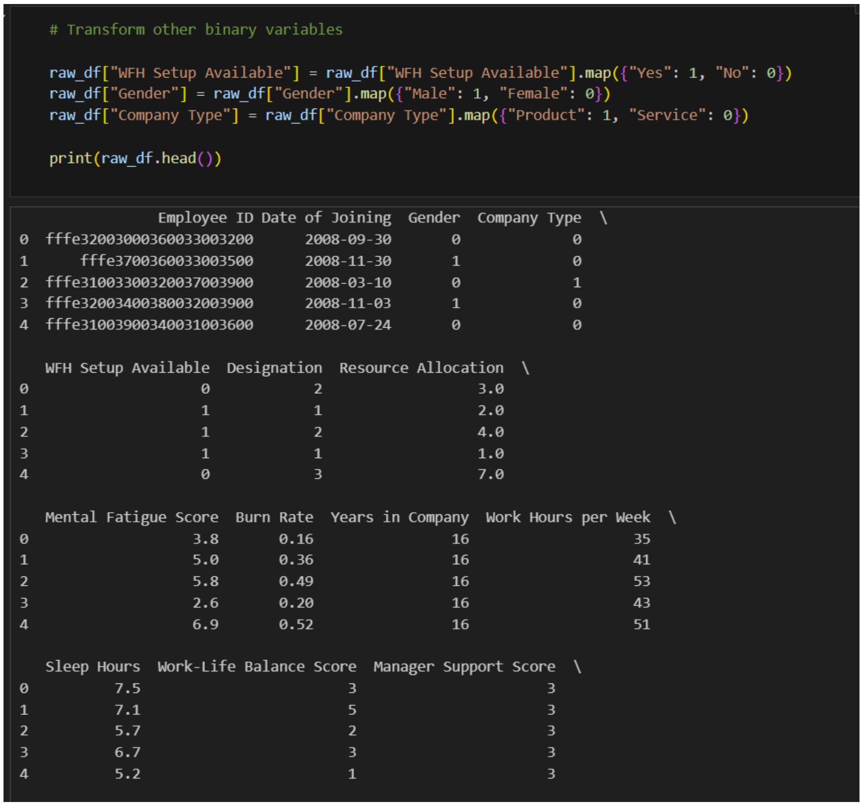
View the distinct categories for “Gender” and “Company Type”:



```
# Distinct Categories for Gender and Company Type

print(raw_df["Gender"].unique())
print(raw_df["Company Type"].unique())
```

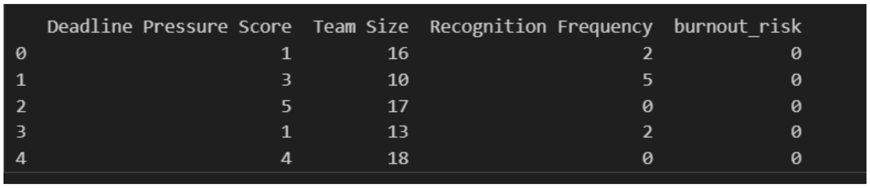
```
['Female' 'Male']
['Service' 'Product']
```

Transform binary variables and view the new cleaned head of the dataframe:



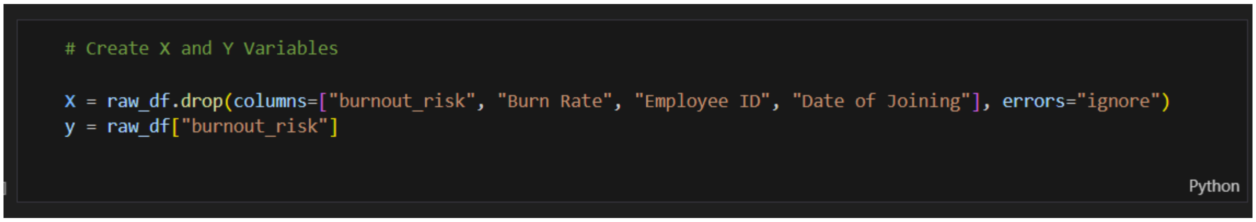
```
# Transform other binary variables

raw_df["WFH Setup Available"] = raw_df["WFH Setup Available"].map({"Yes": 1, "No": 0})
raw_df["Gender"] = raw_df["Gender"].map({"Male": 1, "Female": 0})
raw_df["Company Type"] = raw_df["Company Type"].map({"Product": 1, "Service": 0})

print(raw_df.head())
```

```
                Employee ID Date of Joining  Gender  Company Type  \
0  fffe32003000360033003200      2008-09-30       0             0   
1      fffe3700360033003500      2008-11-30       1             0   
2  fffe31003300320037003900      2008-03-10       0             1   
3  fffe32003400380032003900      2008-11-03       1             0   
4  fffe31003900340031003600      2008-07-24       0             0   

   WFH Setup Available  Designation  Resource Allocation  \
0                    0            2                  3.0   
1                    1            1                  2.0   
2                    1            2                  4.0   
3                    1            1                  1.0   
4                    0            3                  7.0   

   Mental Fatigue Score  Burn Rate  Years in Company  Work Hours per Week  \
0                   3.8       0.16                16                   35   
1                   5.0       0.36                16                   41   
2                   5.8       0.49                16                   53   
3                   2.6       0.20                16                   43   
4                   6.9       0.52                16                   51   

   Sleep Hours  Work-Life Balance Score  Manager Support Score  \
0          7.5                        3                      3   
1          7.1                        5                      3   
2          5.7                        2                      3   
3          6.7                        3                      3   
4          5.2                        1                      3   
```



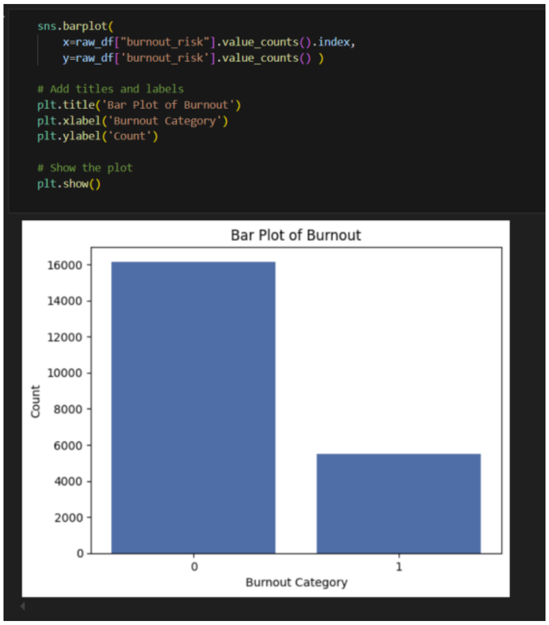
```
   Deadline Pressure Score  Team Size  Recognition Frequency  burnout_risk
0                        1         16                      2             0
1                        3         10                      5             0
2                        5         17                      0             0
3                        1         13                      2             0
4                        4         18                      0             0
```

Once the underlying data of the analysis has been prepared, further data exploration was conducted, and the X (independent) and Y (dependent) variables are defined and examined by use of graphs (univariate and bivariate). Variable X includes all variables except for “burnout_risk”, “Burn Rate”, “Employee ID”, and “Date of Joining”. Burnout_risk shall be the Y variable, and Burn Rate was used to create the Y variable; therefore, it cannot be included in the X variables. “Employee ID” and “Date of Joining” are inconsequential to the goal of this analysis and shall not be included in the X or Y variables.



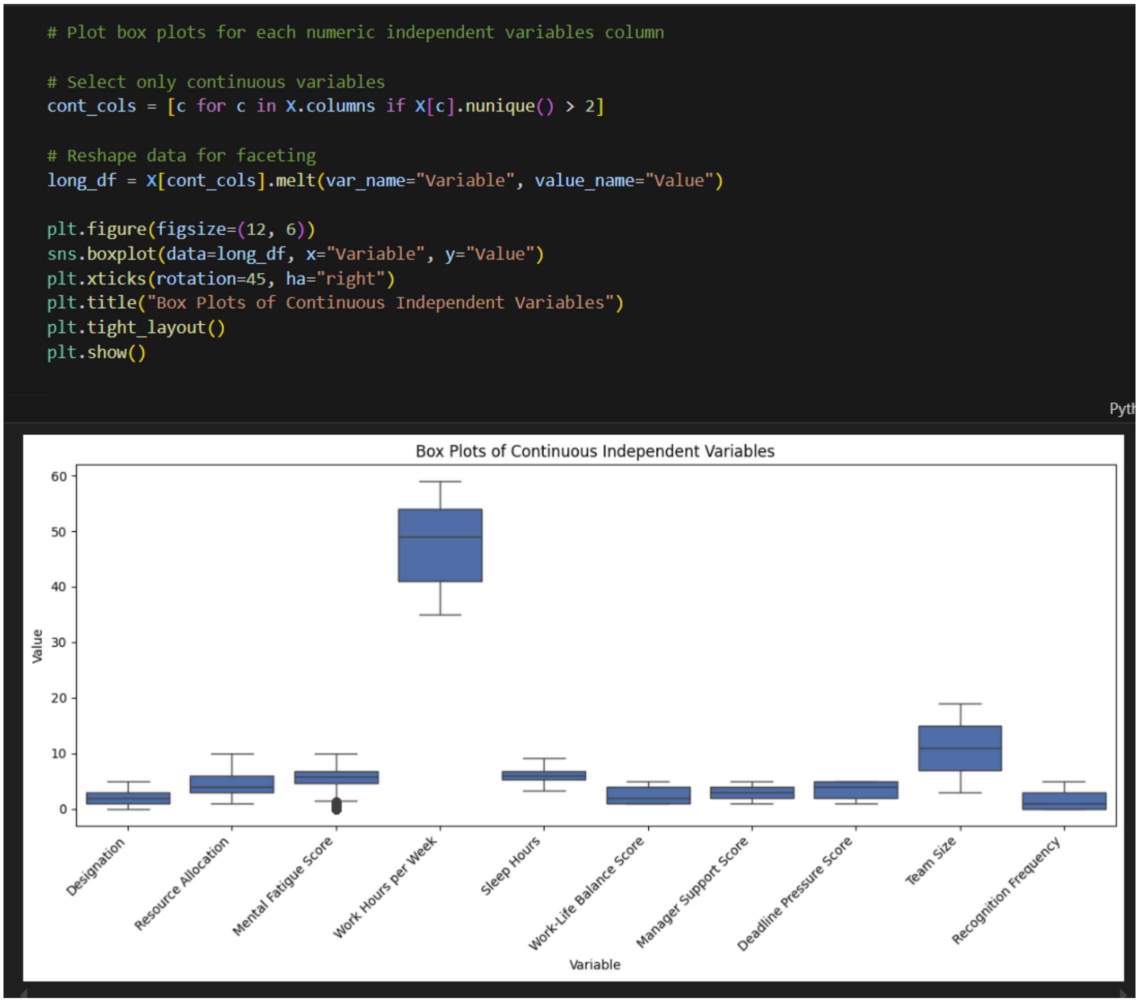
```
# Create X and Y Variables

X = raw_df.drop(columns=["burnout_risk", "Burn Rate", "Employee ID", "Date of Joining"], errors="ignore")
y = raw_df["burnout_risk"]
```

To understand the spread and relationship between variables, a bar graph, a variety of boxplots, and a heatmap were created. The first univariate graph to be generated was a bar graph to visualize the ratio of the dependent variable named burnout_risk.



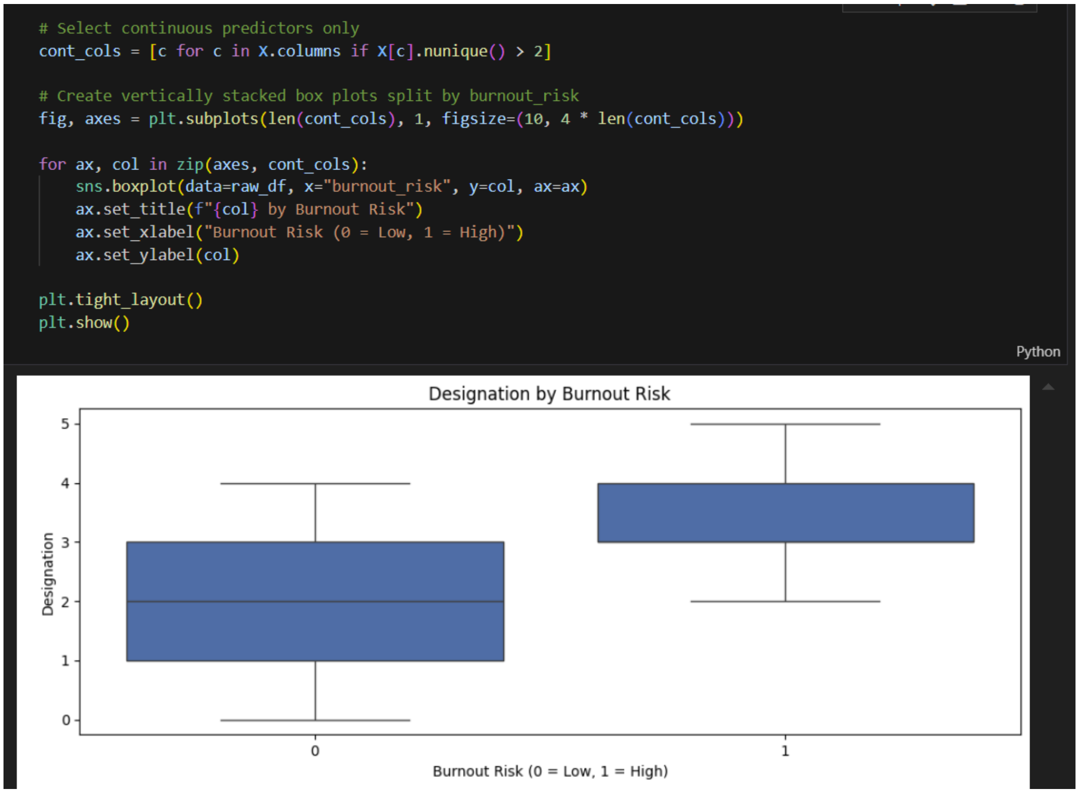
```
sns.barplot(
    x=raw_df["burnout_risk"].value_counts().index,
    y=raw_df['burnout_risk'].value_counts() )

# Add titles and labels
plt.title('Bar Plot of Burnout')
plt.xlabel('Burnout Category')
plt.ylabel('Count')

# Show the plot
plt.show()
```

The next was a combination of boxplots for all numeric independent variables to examine for outliers and distributions.

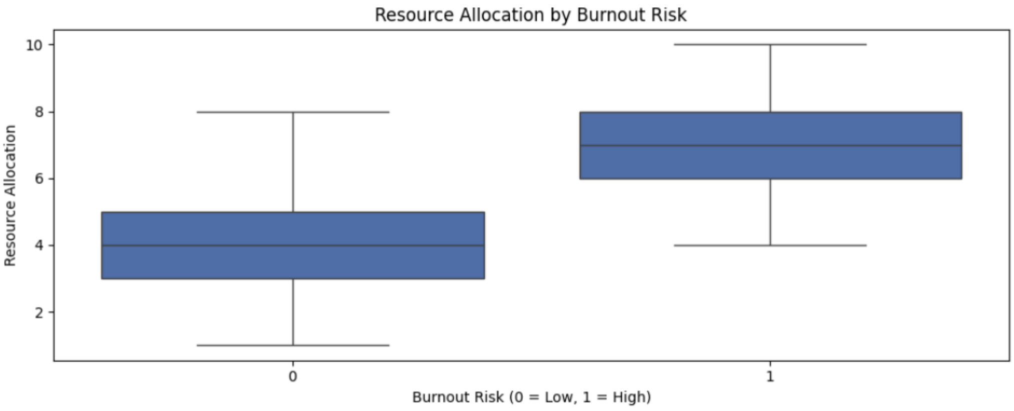


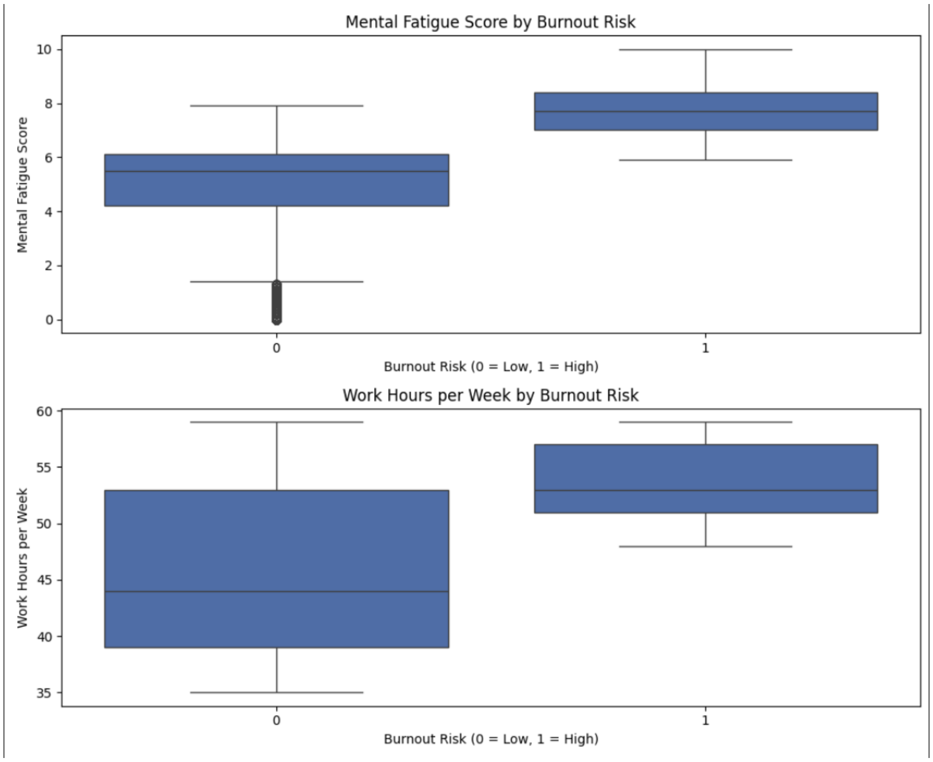
```python
# Plot box plots for each numeric independent variables column

# Select only continuous variables
cont_cols = [c for c in X.columns if X[c].nunique() > 2]

# Reshape data for faceting
long_df = X[cont_cols].melt(var_name="Variable", value_name="Value")

plt.figure(figsize=(12, 6))
sns.boxplot(data=long_df, x="Variable", y="Value")
plt.xticks(rotation=45, ha="right")
plt.title("Box Plots of Continuous Independent Variables")
plt.tight_layout()
plt.show()
```

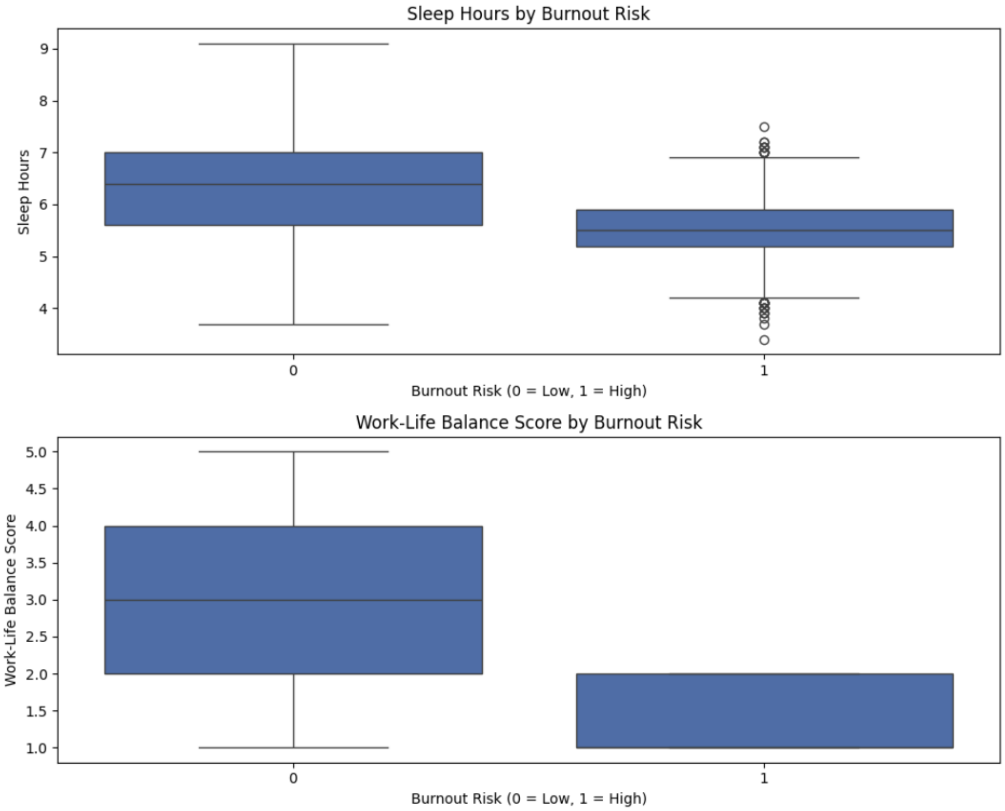
Then, the independent variable boxplots were split by burnout_risk to explore and visualize differences in variable distribution between employees classified as high and low risk of Burnout. Several variables exhibited noticeable shifts in median values and dispersion across outcome groups, suggesting potential associations with burnout risk.

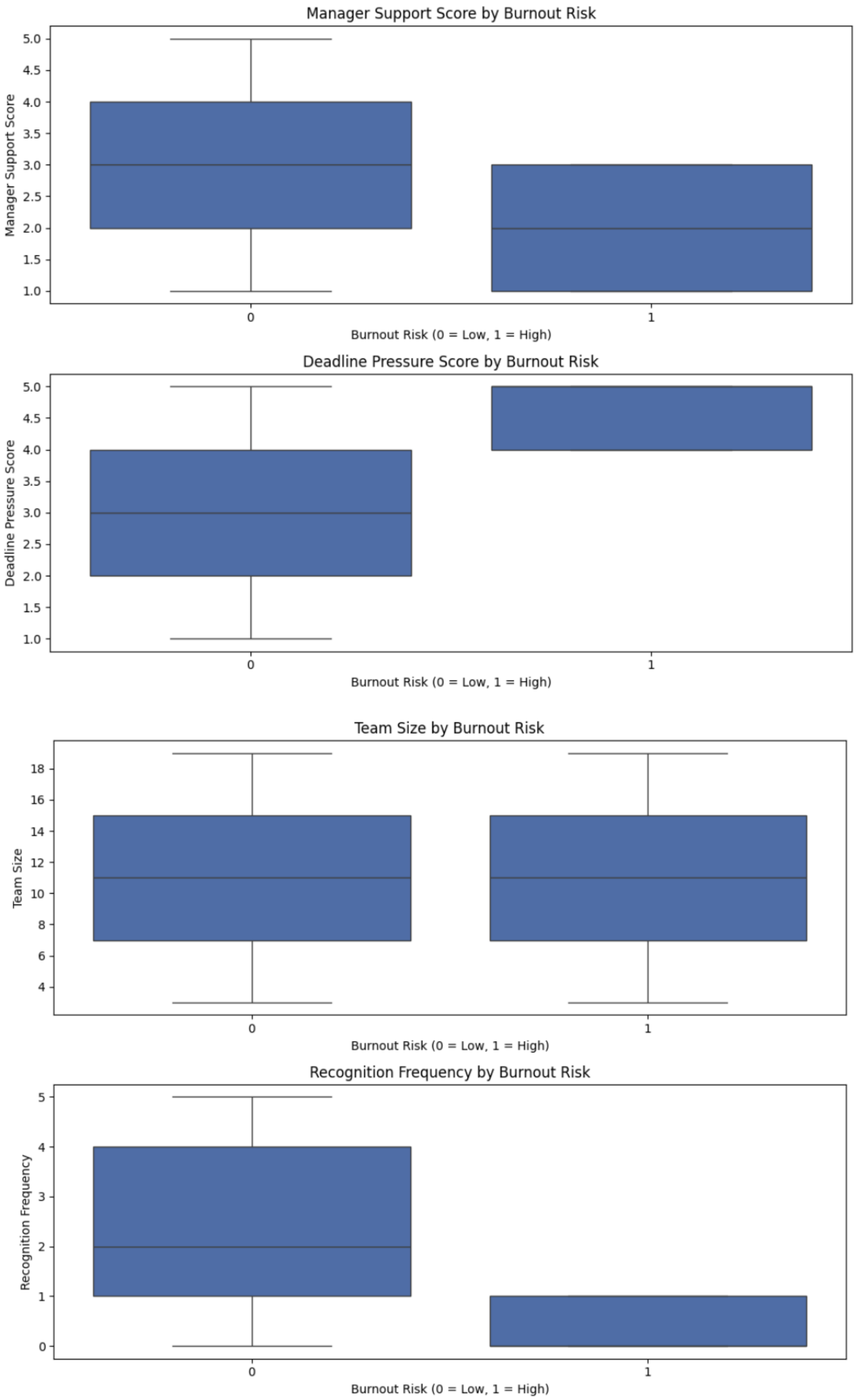


```python
# Select continuous predictors only
cont_cols = [c for c in X.columns if X[c].nunique() > 2]

# Create vertically stacked box plots split by burnout_risk
fig, axes = plt.subplots(len(cont_cols), 1, figsize=(10, 4 * len(cont_cols)))

for ax, col in zip(axes, cont_cols):
    sns.boxplot(data=raw_df, x="burnout_risk", y=col, ax=ax)
    ax.set_title(f"{col} by Burnout Risk")
    ax.set_xlabel("Burnout Risk (0 = Low, 1 = High)")
    ax.set_ylabel(col)

plt.tight_layout()
plt.show()
```

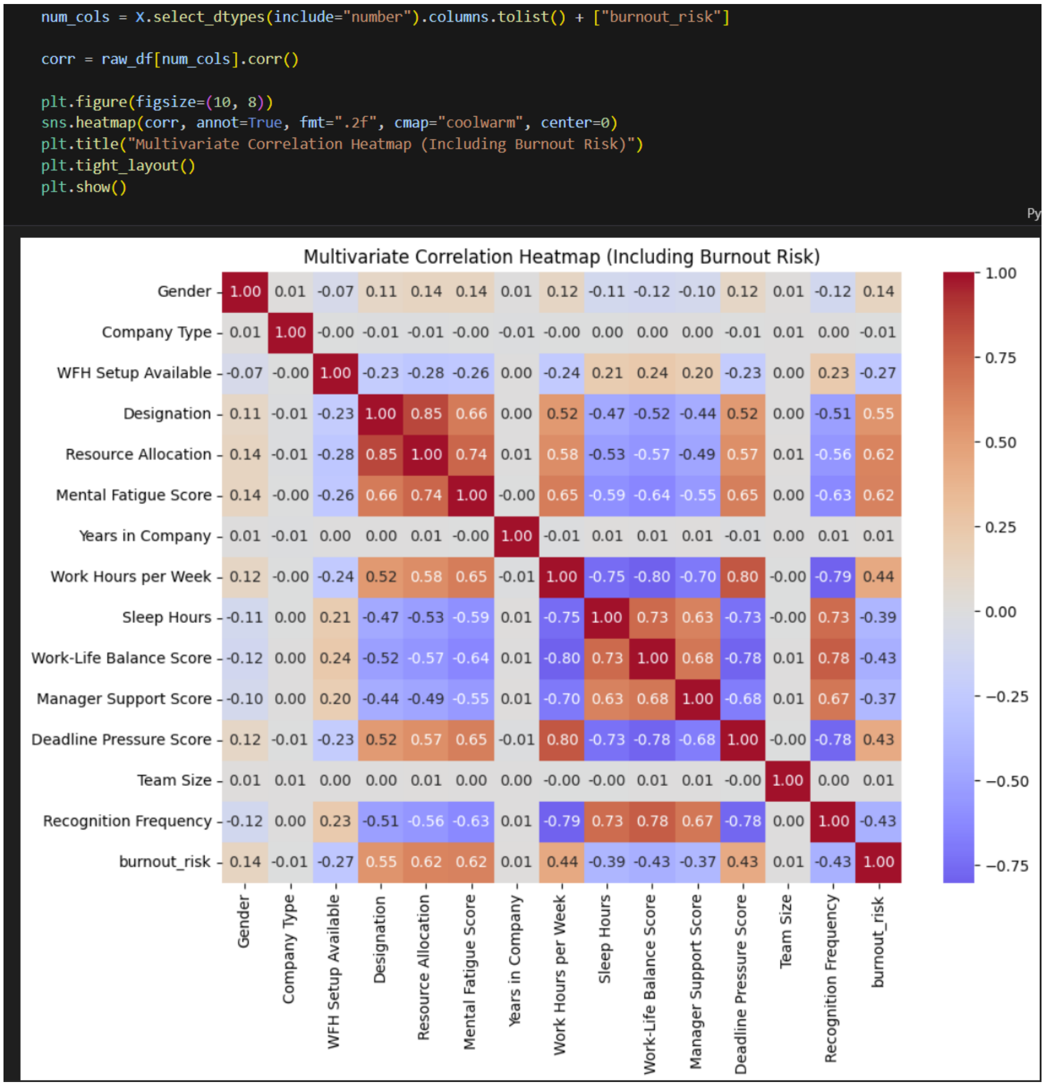



Mental Fatigue Score by Burnout Risk
Mental Fatigue Score
0
2
4
6
8
10
0
1
Burnout Risk (0 = Low, 1 = High)
Work Hours per Week by Burnout Risk
Work Hours per Week
35
40
45
50
55
60
0
1
Burnout Risk (0 = Low, 1 = High)


Sleep Hours by Burnout Risk
Sleep Hours
4
5
6
7
8
9
0
1
Burnout Risk (0 = Low, 1 = High)
Work-Life Balance Score by Burnout Risk
Work-Life Balance Score
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
0
1
Burnout Risk (0 = Low, 1 = High)



The final visualization created was a Multivariate Correlation Heatmap (including Burnout Risk). This heatmap assesses the relationship between numeric predictor variables and identifies potential multicollinearity prior to model development.



```
num_cols = X.select_dtypes(include="number").columns.tolist() + ["burnout_risk"]

corr = raw_df[num_cols].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap="coolwarm", center=0)
plt.title("Multivariate Correlation Heatmap (Including Burnout Risk)")
plt.tight_layout()
plt.show()
```

Multivariate Correlation Heatmap (Including Burnout Risk)

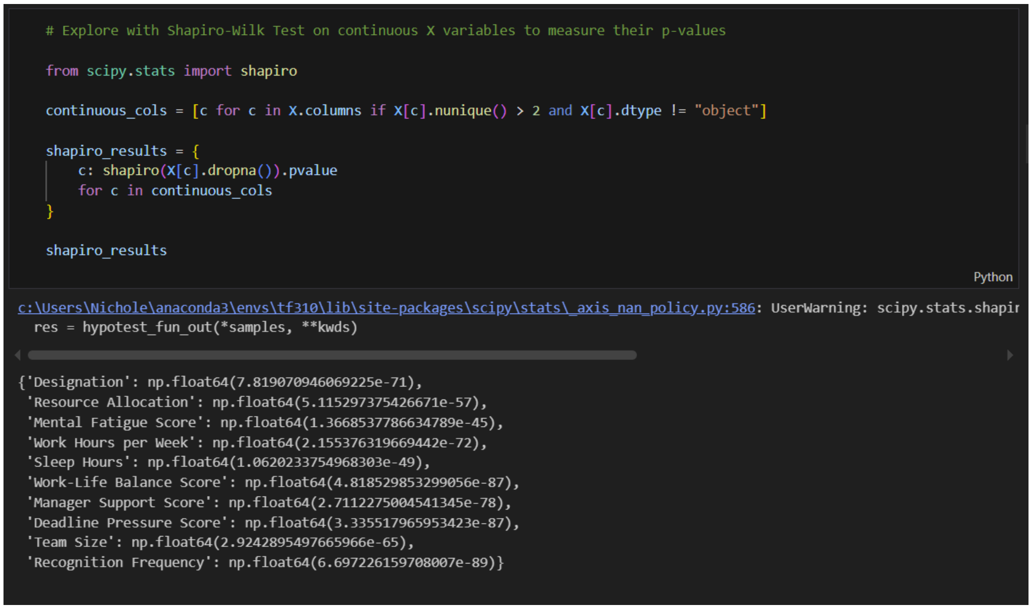
	Gender	Company Type	WFH Setup Available	Designation	Resource Allocation	Mental Fatigue Score	Years in Company	Work Hours per Week	Sleep Hours	Work-Life Balance Score	Manager Support Score	Deadline Pressure Score	Team Size	Recognition Frequency	burnout_risk
Gender	1.00	0.01	-0.07	0.11	0.14	0.14	0.01	0.12	-0.11	-0.12	-0.10	0.12	0.01	-0.12	0.14
Company Type	0.01	1.00	-0.00	-0.01	-0.01	-0.00	-0.01	-0.00	0.00	0.00	0.00	-0.01	0.01	0.00	-0.01
WFH Setup Available	-0.07	-0.00	1.00	-0.23	-0.28	-0.26	0.00	-0.24	0.21	0.24	0.20	-0.23	0.00	0.23	-0.27
Designation	0.11	-0.01	-0.23	1.00	0.85	0.66	0.00	0.52	-0.47	-0.52	-0.44	0.52	0.00	-0.51	0.55
Resource Allocation	0.14	-0.01	-0.28	0.85	1.00	0.74	0.01	0.58	-0.53	-0.57	-0.49	0.57	0.01	-0.56	0.62
Mental Fatigue Score	0.14	-0.00	-0.26	0.66	0.74	1.00	-0.00	0.65	-0.59	-0.64	-0.55	0.65	0.00	-0.63	0.62
Years in Company	0.01	-0.01	0.00	0.00	0.01	-0.00	1.00	-0.01	0.01	0.01	0.01	-0.01	0.00	0.01	0.01
Work Hours per Week	0.12	-0.00	-0.24	0.52	0.58	0.65	-0.01	1.00	-0.75	-0.80	-0.70	0.80	-0.00	-0.79	0.44
Sleep Hours	-0.11	0.00	0.21	-0.47	-0.53	-0.59	0.01	-0.75	1.00	0.73	0.63	-0.73	-0.00	0.73	-0.39
Work-Life Balance Score	-0.12	0.00	0.24	-0.52	-0.57	-0.64	0.01	-0.80	0.73	1.00	0.68	-0.78	0.01	0.78	-0.43
Manager Support Score	-0.10	0.00	0.20	-0.44	-0.49	-0.55	0.01	-0.70	0.63	0.68	1.00	-0.68	0.01	0.67	-0.37
Deadline Pressure Score	0.12	-0.01	-0.23	0.52	0.57	0.65	-0.01	0.80	-0.73	-0.78	-0.68	1.00	-0.00	-0.78	0.43
Team Size	0.01	0.01	0.00	0.00	0.01	0.00	0.00	-0.00	-0.00	0.01	0.01	-0.00	1.00	0.00	0.01
Recognition Frequency	-0.12	0.00	0.23	-0.51	-0.56	-0.63	0.01	-0.79	0.73	0.78	0.67	-0.78	0.00	1.00	-0.43
burnout_risk	0.14	-0.01	-0.27	0.55	0.62	0.62	0.01	0.44	-0.39	-0.43	-0.37	0.43	0.01	-0.43	1.00

The data extraction and preparation methods used in this study were appropriate for the structure and scope of the dataset to produce a clean, analysis-ready dataset suitable for binary classification modeling. The final prepared dataset consists of 21,626 observations and 15 variables (including the dependent and independent variables).

Section D: Analysis

The primary analytical technique used in this study was binary logistic regression. Logistic regression is appropriate when the dependent variable represents a binary outcome, in this case, whether an employee is classified as being high or low risk of Burnout. This technique estimates the probability that an observation belongs to a class by modeling the relationship between variables and the log-odds of the outcome. One advantage of logistic regression when predicting burnout risk is that the results are interpretable with the binary outcome of 1 or 0. The disadvantage of logistic regression is that it assumes a linear relationship between Burnout and the other variables.

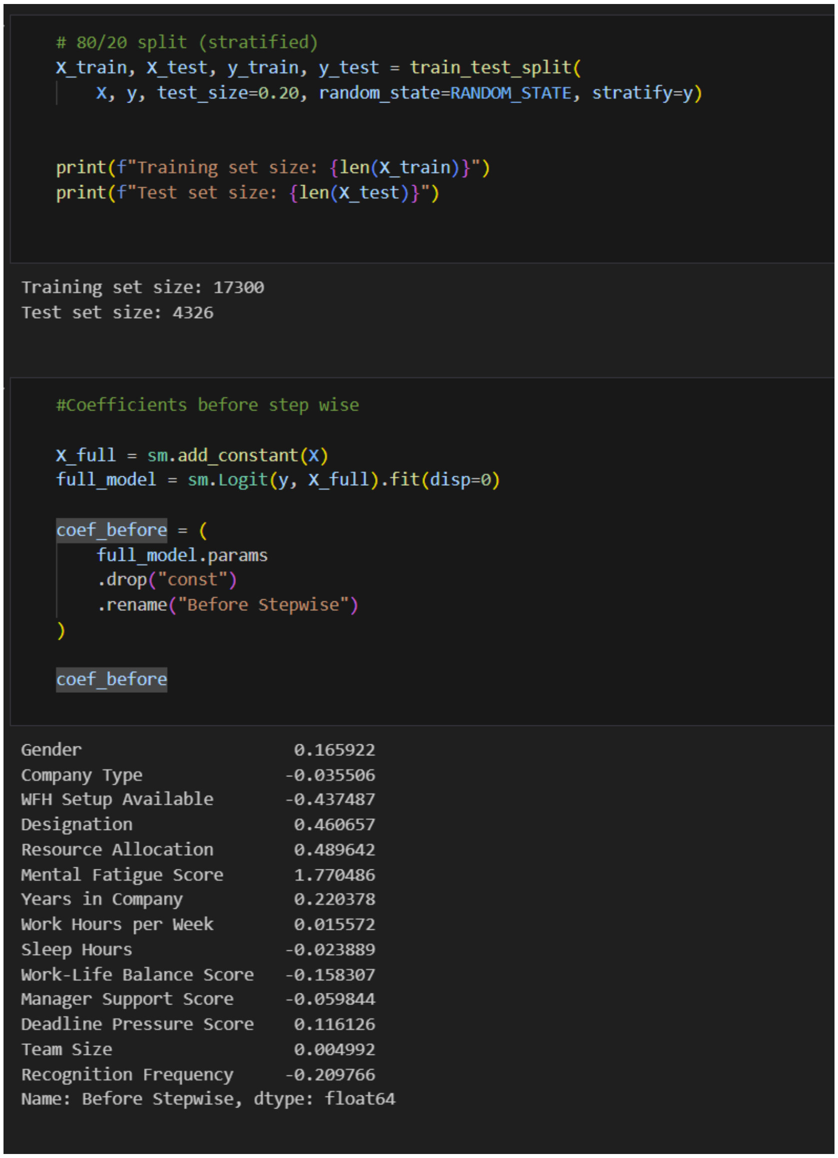
Although logistic regression does not require predictor variables to be normally distributed, the Shapiro–Wilk test was applied to continuous predictors as an exploratory step prior to model development. The Shapiro–Wilk test assesses whether a variable deviates from a normal distribution and provides additional insight into the distributional characteristics of the data. Results indicated that ten continuous variables deviated from normality, which is consistent with expectations for large datasets. While the Shapiro–Wilk test is commonly applied to smaller sample sizes and can be overly sensitive in large datasets, it was not used as a determinant for model selection. Instead, its results were interpreted in conjunction with visual exploratory methods, including histograms, box plots, and correlation analysis. This combined approach supported informed data preparation decisions and confirmed that deviations from normality did not violate the assumptions of logistic regression. The image below displays the test results.



```
# Explore with Shapiro-Wilk Test on continuous X variables to measure their p-values

from scipy.stats import shapiro

continuous_cols = [c for c in X.columns if X[c].nunique() > 2 and X[c].dtype != "object"]

shapiro_results = {
    c: shapiro(X[c].dropna()).pvalue
    for c in continuous_cols
}

shapiro_results
```

```
c:\Users\Nichole\anaconda3\envs\tf310\lib\site-packages\scipy\stats\_axis_nan_policy.py:586: UserWarning: scipy.stats.shapir
  res = hypotest_fun_out(*samples, **kwds)

{'Designation': np.float64(7.819070946069225e-71),
 'Resource Allocation': np.float64(5.115297375426671e-57),
 'Mental Fatigue Score': np.float64(1.3668537786634789e-45),
 'Work Hours per Week': np.float64(2.155376319669442e-72),
 'Sleep Hours': np.float64(1.0620233754968303e-49),
 'Work-Life Balance Score': np.float64(4.818529853299056e-87),
 'Manager Support Score': np.float64(2.7112275004541345e-78),
 'Deadline Pressure Score': np.float64(3.335517965953423e-87),
 'Team Size': np.float64(2.9242895497665966e-65),
 'Recognition Frequency': np.float64(6.697226159708007e-89)}
```

Following exploratory analysis, the dataset was split into training and testing subsets for model evaluation. The logistic regression model used X variables as predictors for the target outcome, Y variable. Model coefficients were generated to identify parameters that may influence the probability of the 0 or 1 outcome.



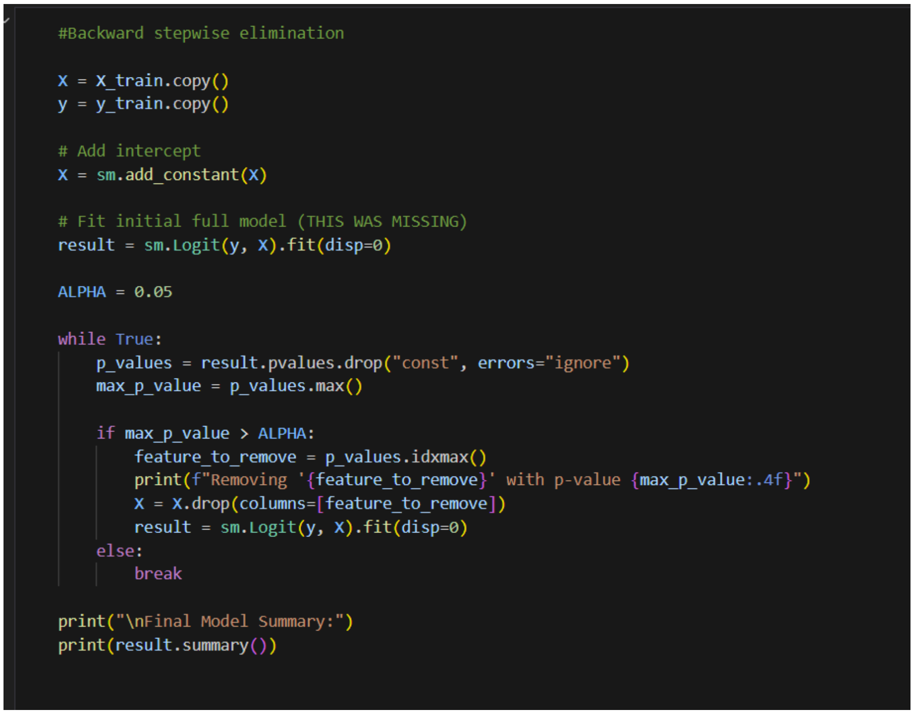
```
# 80/20 split (stratified)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=RANDOM_STATE, stratify=y)


print(f"Training set size: {len(X_train)}")
print(f"Test set size: {len(X_test)}")
```

```
Training set size: 17300
Test set size: 4326
```

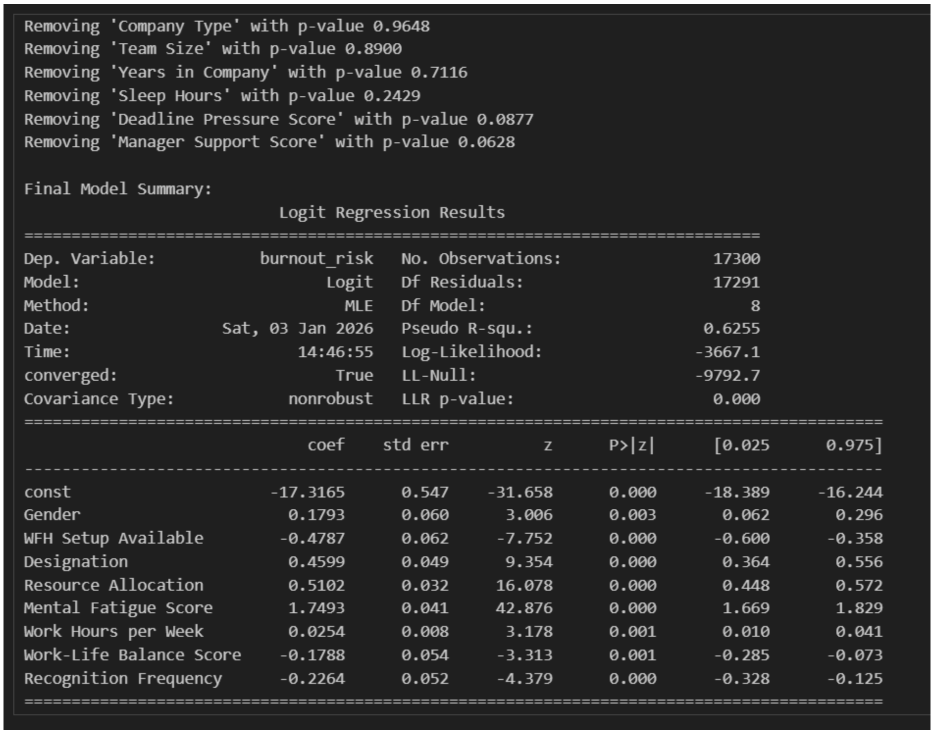
```
#Coefficients before step wise

X_full = sm.add_constant(X)
full_model = sm.Logit(y, X_full).fit(disp=0)

coef_before = (
    full_model.params
    .drop("const")
    .rename("Before Stepwise")
)

coef_before
```

```
Gender                      0.165922
Company Type               -0.035506
WFH Setup Available        -0.437487
Designation                 0.460657
Resource Allocation         0.489642
Mental Fatigue Score        1.770486
Years in Company            0.220378
Work Hours per Week         0.015572
Sleep Hours                -0.023889
Work-Life Balance Score    -0.158307
Manager Support Score      -0.059844
Deadline Pressure Score     0.116126
Team Size                   0.004992
Recognition Frequency      -0.209766
Name: Before Stepwise, dtype: float64
```

To refine the model and identify the best independent variables, backward stepwise elimination was applied. This technique involves iteratively removing predictor variables that do not contribute to the model, based on statistical significance and model fit, when the p-value is greater than 0.05. At each step, the least significant variable is removed, and the model is refitted. This process is repeated until only meaningful variables are left. The advantage of this technique is that it reduces potential multicollinearity and overfitting. The results for this technique are provided in two screenshots below, one of the code and the other of the output.

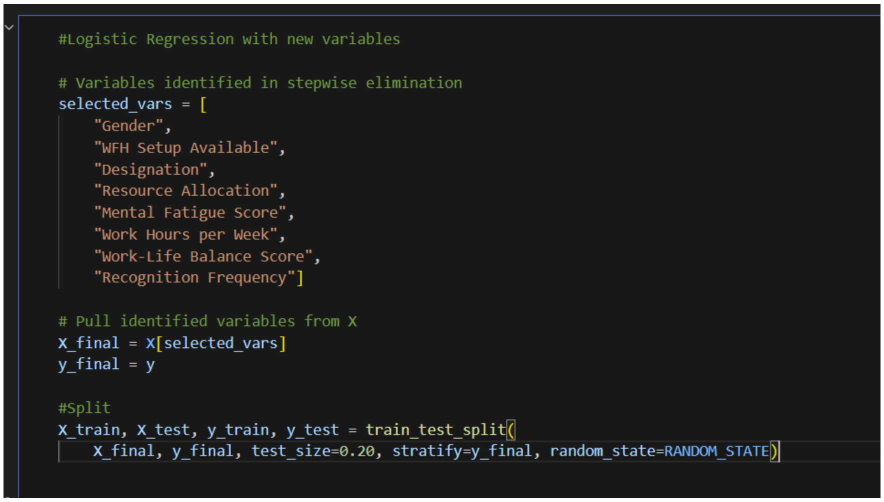


```python
#Backward stepwise elimination

X = X_train.copy()
y = y_train.copy()

# Add intercept
X = sm.add_constant(X)

# Fit initial full model (THIS WAS MISSING)
result = sm.Logit(y, X).fit(disp=0)

ALPHA = 0.05

while True:
    p_values = result.pvalues.drop("const", errors="ignore")
    max_p_value = p_values.max()

    if max_p_value > ALPHA:
        feature_to_remove = p_values.idxmax()
        print(f"Removing '{feature_to_remove}' with p-value {max_p_value:.4f}")
        X = X.drop(columns=[feature_to_remove])
        result = sm.Logit(y, X).fit(disp=0)
    else:
        break

print("\nFinal Model Summary:")
print(result.summary())
```



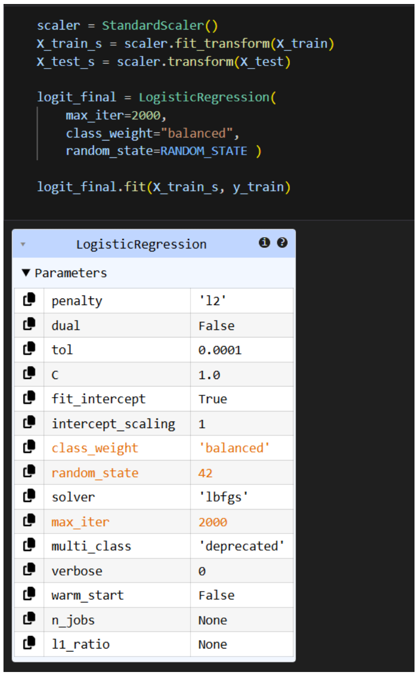
```
Removing 'Company Type' with p-value 0.9648
Removing 'Team Size' with p-value 0.8900
Removing 'Years in Company' with p-value 0.7116
Removing 'Sleep Hours' with p-value 0.2429
Removing 'Deadline Pressure Score' with p-value 0.0877
Removing 'Manager Support Score' with p-value 0.0628

Final Model Summary:
                           Logit Regression Results
==============================================================================
Dep. Variable:            burnout_risk   No. Observations:                17300
Model:                           Logit   Df Residuals:                    17291
Method:                            MLE   Df Model:                            8
Date:                 Sat, 03 Jan 2026   Pseudo R-squ.:                  0.6255
Time:                         14:46:55   Log-Likelihood:                -3667.1
converged:                        True   LL-Null:                       -9792.7
Covariance Type:             nonrobust   LLR p-value:                     0.000
=============================================================================================
                              coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------
const                     -17.3165      0.547    -31.658      0.000     -18.389     -16.244
Gender                      0.1793      0.060      3.006      0.003       0.062       0.296
WFH Setup Available        -0.4787      0.062     -7.752      0.000      -0.600      -0.358
Designation                 0.4599      0.049      9.354      0.000       0.364       0.556
Resource Allocation         0.5102      0.032     16.078      0.000       0.448       0.572
Mental Fatigue Score        1.7493      0.041     42.876      0.000       1.669       1.829
Work Hours per Week         0.0254      0.008      3.178      0.001       0.010       0.041
Work-Life Balance Score    -0.1788      0.054     -3.313      0.001      -0.285      -0.073
Recognition Frequency      -0.2264      0.052     -4.379      0.000      -0.328      -0.125
=============================================================================================
```

The variables that will be retained for the final logistic regression model are "Gender", "WFH Setup Available", "Designation", "Resource Allocation", "Mental Fatigue Score", "Work Hours per Week", "Work-Life Balance Score", and "Recognition Frequency". X was redefined using the statistically significant variables, and the dataset was split into training and testing subsets for the final logistic regression. The training set contained 80% of the records, while the testing set contained 20%. The image below reflects the code used to achieve this step in the analysis.



```
#Logistic Regression with new variables

# Variables identified in stepwise elimination
selected_vars = [
    "Gender",
    "WFH Setup Available",
    "Designation",
    "Resource Allocation",
    "Mental Fatigue Score",
    "Work Hours per Week",
    "Work-Life Balance Score",
    "Recognition Frequency"]

# Pull identified variables from X
X_final = X[selected_vars]
y_final = y

#Split
X_train, X_test, y_train, y_test = train_test_split(
    X_final, y_final, test_size=0.20, stratify=y_final, random_state=RANDOM_STATE)
```

Prior to the final model fitting, the independent variables (X) were stabilized using StandardScaler to ensure that variables measured on different scales contributed proportionally to the model. Standardization was applied to the training and testing data using the same fitted scaler. The logistic regression model was configured with L2 regularization and optimized using the limited memory Broyden Felcher Goldfarb Shanno (LBFGS) solver. A maximum iteration of 2,000 was specified to support the model. Class weights were set to "balanced" to account for potential imbalance.



```
scaler = StandardScaler()
X_train_s = scaler.fit_transform(X_train)
X_test_s = scaler.transform(X_test)

logit_final = LogisticRegression(
    max_iter=2000,
    class_weight="balanced",
    random_state=RANDOM_STATE )

logit_final.fit(X_train_s, y_train)
```

LogisticRegression

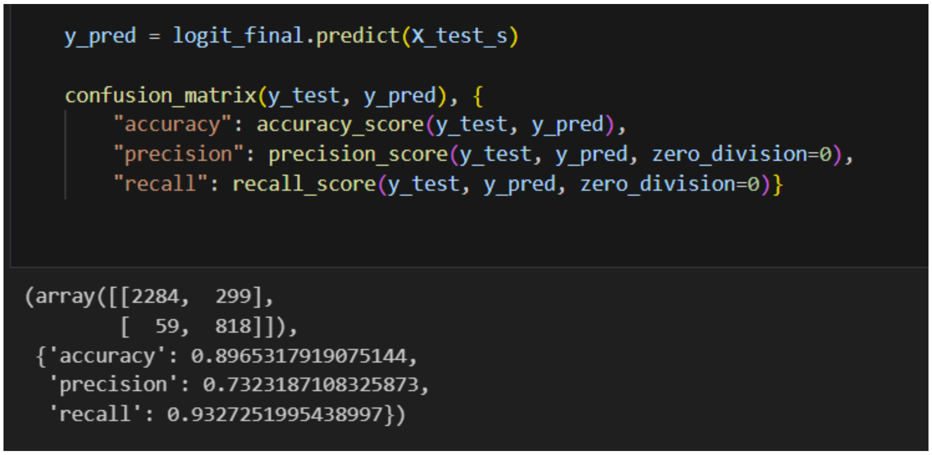
▼ Parameters

penalty	'l2'
dual	False
tol	0.0001
C	1.0
fit_intercept	True
intercept_scaling	1
class_weight	'balanced'
random_state	42
solver	'lbfgs'
max_iter	2000
multi_class	'deprecated'
verbose	0
warm_start	False
n_jobs	None
l1_ratio	None

Once the final model was created, performance metrics were reviewed, including classification accuracy, precision score, and recall, and a confusion matrix was generated. The model achieved an accuracy of 89.65%, a precision score of 73.23%, and a recall score of 93.27%.

The results of the confusion matrix show that the model correctly classified 2,284 true negative observations, representing employees accurately as being at low risk of Burnout, and 818 true positive observations, representing employees correctly classified as being at high risk for Burnout. The model produced 299 false positive predictions, where employees were classified as high risk despite being actually low risk, and 59 false negative predictions, where high-risk employees were incorrectly classified as low risk.

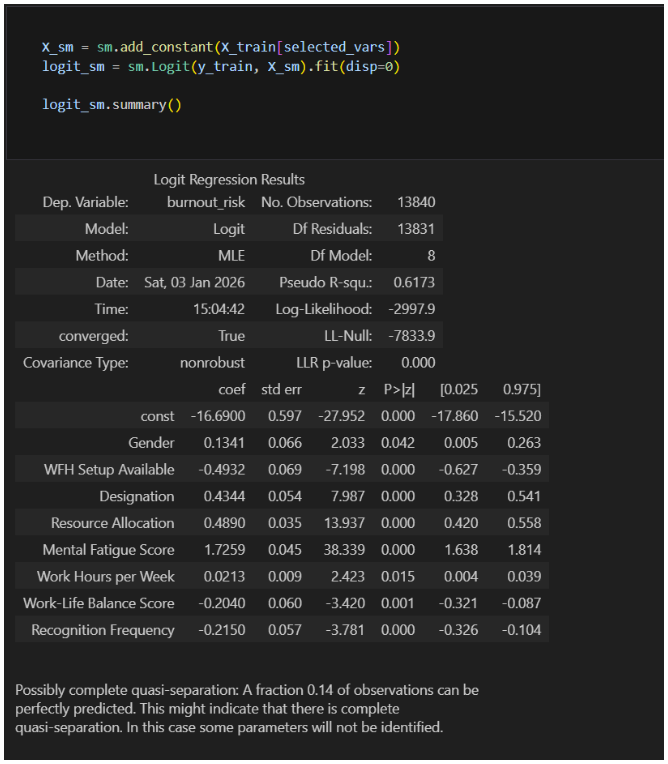
The high recall score indicates that the model was effective at identifying employees at high risk of Burnout. The lower precision score indicates that the model finds employees at high risk of Burnout, but could also classify some low-risk employees as high-risk.



```
y_pred = logit_final.predict(X_test_s)

confusion_matrix(y_test, y_pred), {
    "accuracy": accuracy_score(y_test, y_pred),
    "precision": precision_score(y_test, y_pred, zero_division=0),
    "recall": recall_score(y_test, y_pred, zero_division=0)}
```

```
(array([[2284,  299],
        [  59,  818]]),
 {'accuracy': 0.8965317919075144,
  'precision': 0.7323187108325873,
  'recall': 0.9327251995438997})
```

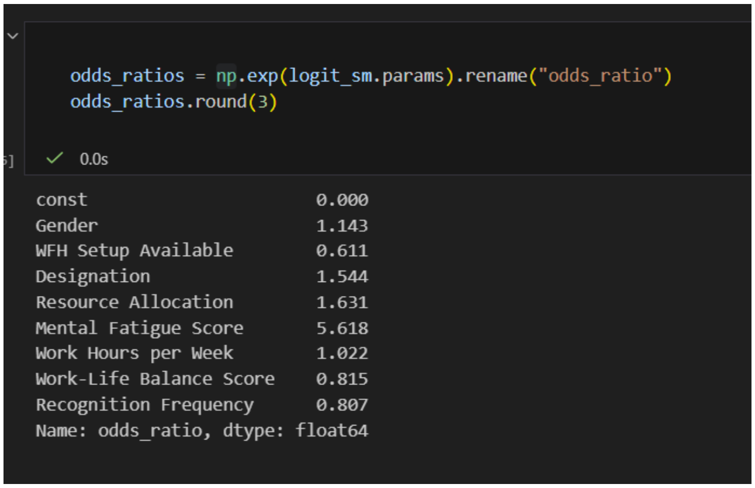
Next, a logistic regression model was fitted using the Statsmodels Logit function on retained variables. The model's pseudo R-squared value of 0.62 indicates strong explanatory power for a logistic regression model. All retained predictors have a statistically significant p-value of less than 0.05. Gender, Designation, Resource Allocation, Mental Fatigue Score, and Work Hours per Week have a positive association with Burnout. WFH Setup Available, Work-life Balance Score, and Recognition Frequency have a negative association with Burnout.



```python
X_sm = sm.add_constant(X_train[selected_vars])
logit_sm = sm.Logit(y_train, X_sm).fit(disp=0)

logit_sm.summary()
```

```
                         Logit Regression Results
   Dep. Variable:      burnout_risk   No. Observations:   13840
           Model:             Logit       Df Residuals:   13831
          Method:               MLE           Df Model:       8
            Date:  Sat, 03 Jan 2026      Pseudo R-squ.:  0.6173
            Time:          15:04:42     Log-Likelihood:  -2997.9
       converged:              True            LL-Null:  -7833.9
 Covariance Type:         nonrobust        LLR p-value:   0.000
                                coef  std err        z  P>|z|   [0.025   0.975]
                     const  -16.6900    0.597  -27.952  0.000  -17.860  -15.520
                    Gender    0.1341    0.066    2.033  0.042    0.005    0.263
        WFH Setup Available  -0.4932    0.069   -7.198  0.000   -0.627   -0.359
                Designation   0.4344    0.054    7.987  0.000    0.328    0.541
        Resource Allocation   0.4890    0.035   13.937  0.000    0.420    0.558
        Mental Fatigue Score  1.7259    0.045   38.339  0.000    1.638    1.814
         Work Hours per Week  0.0213    0.009    2.423  0.015    0.004    0.039
     Work-Life Balance Score -0.2040    0.060   -3.420  0.001   -0.321   -0.087
       Recognition Frequency -0.2150    0.057   -3.781  0.000   -0.326   -0.104

Possibly complete quasi-separation: A fraction 0.14 of observations can be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

Logit summary Odds Ratios were examined to understand how changes in the variables affect the likelihood of high burnout classification. For example, the Mental Fatigue Score is the most dominant influencer of an increased likelihood of high-risk burnout classification.



```python
odds_ratios = np.exp(logit_sm.params).rename("odds_ratio")
odds_ratios.round(3)
```

```
✓ 0.0s
```

```
const                      0.000
Gender                     1.143
WFH Setup Available        0.611
Designation                1.544
Resource Allocation        1.631
Mental Fatigue Score       5.618
Work Hours per Week        1.022
Work-Life Balance Score    0.815
Recognition Frequency      0.807
Name: odds_ratio, dtype: float64
```

Overall, the steps and techniques utilized in the data analysis of this study, including exploratory diagnostics, backward stepwise variable selection, and logistic regression modeling, were suitable for the data structure and the study's goal. The methods discussed above provided analytical support to classify employees' risk of Burnout based on the workplace. The summary and implications shall be discussed next.

Section E. Data Summary and Implications

Results of the data analysis indicate that workplace data can be used to classify employees as being at high or low risk of Burnout. The logistic regression model demonstrated strong predictive performance, achieving an accuracy of 89.65%. Based on this result, the alternative hypothesis is accepted, a logistic regression model can predict whether employees are at a high or low risk of Burnout with an accuracy greater than 70%.

Analysis outputs identified several variables that were associated with an increased risk of Burnout. Mental fatigue score, resource allocation, designation, work hours per week, and gender were positively associated with an increased risk of burnout. WFH Setup Available, work-life balance score, and recognition frequency lower the risk of Burnout.

One limitation of the analysis conducted is the use of a quartile-based threshold to define the binary burnout indicator based on the “Burn Rate”. This approach focused on employees with the highest level of Burnout, which may be an oversimplification of a complex topic like employee well-being.

A course of action based on the outcome of this study would be to identify employees at risk of Burnout and implement a form of intervention to attempt to lower the burnout risk. An intervention could include interviewing employees with a high level of Burnout to attempt to find the root cause of mental fatigue, since it is the highest contributor to burnout risk.

Future research of this dataset includes alternative modeling of the original sourced data, such as an ensemble method to capture potential nonlinear relationships amongst workplace variables. Another direction for this study would be to recreate the survey questionnaire, send it to employees every six months, and incorporate a time series study to track employee well-being and Burnout over a period.

In conclusion, the outcome of this study creates a logistic regression predictive modeling tool that can help organizations be proactive in identifying employees at high risk of Burnout. While the findings are subject to limitations, the data analysis highlights the potential value of a data-driven approach to promoting employee well-being and supports informed decision-making by an organization.

Section F. Sources

Devkar, P. (2025). *Burnout Among Corporate Employees*. Harvard Dataverse. Dataset retrieved December 10, 2025, from https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/VG6KQD

Liu, C., Chuang, Y.-C., Qin, L., Ren, L., Chien, C.-W., & Tung, T.-H. (2025). *Machine Learning Based Model for Analysing and Accurately Predicting Factors Related to Burnout in Healthcare Workers*. BMJ Public Health, 3(2), e000777. https://pmc.ncbi.nlm.nih.gov/articles/PMC12414203/

National Alliance on Mental Illness (NAMI). (2024). *The 2024 NAMI Workplace Mental Health Poll*. NAMI. https://www.nami.org/support-education/publications-reports/survey-reports/the-2024-nami-workplace-mental-health-poll/

Robinson, B. (2025). *Job Burnout at 66% in 2025, New Study Shows.* Forbes. https://www.forbes.com/sites/bryanrobinson/2025/02/08/job-burnout-at-66-in-2025-new-study-shows/

Samal. (2021). *Understanding the Correct Techniques for Missing Value Imputation in Data Science*. Medium. https://medium.com/@abhinandan198/understanding-the-correct-techniques-for-missing-value-imputation-in-data-science-5d9c990494e7

Section G. Professional Communication

To be evaluated based on the above text.