```python
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
```

```python
data = pd.read_csv("Wellbeing_and_lifestyle_data_Kaggle.csv")
data.head()
```

```
  Timestamp  FRUITS_VEGGIES DAILY_STRESS  PLACES_VISITED  CORE_CIRCLE  \
0    7/7/15               3            2               2            5
1    7/7/15               2            3               4            3
2    7/7/15               2            3               3            4
3    7/7/15               3            3              10            3
4    7/7/15               5            1               3            3

   SUPPORTING_OTHERS  SOCIAL_NETWORK  ACHIEVEMENT  DONATION  BMI_RANGE  ...  \
0                  0               5            2         0          1  ...
1                  8              10            5         2          2  ...
2                  4              10            3         2          2  ...
3                 10               7            2         5          2  ...
4                 10               4            2         4          2  ...

   SLEEP_HOURS  LOST_VACATION  DAILY_SHOUTING  SUFFICIENT_INCOME  \
0            7              5               5                  1
1            8              2               2                  2
2            8             10               2                  2
3            5              7               5                  1
4            7              0               0                  2

   PERSONAL_AWARDS  TIME_FOR_PASSION  WEEKLY_MEDITATION         AGE  GENDER  \
0                4                 0                  5    36 to 50  Female
1                3                 2                  6    36 to 50  Female
2                4                 8                  3    36 to 50  Female
3                5                 2                  0  51 or more  Female
4                8                 1                  5  51 or more  Female

   WORK_LIFE_BALANCE_SCORE
0                    609.5
1                    655.6
2                    631.6
3                    622.7
4                    663.9

[5 rows x 24 columns]
```

```python
# Cleaning the data. Do not rerun unless you reload the original dataset first
```

```python
#Timestamp variable
# data['Timestamp'] = pd.to_datetime(data['Timestamp'], errors='coerce')
# data['Timestamp'] = data['Timestamp'].dt.strftime('%m/%d/%Y')
data.drop(["Timestamp"], axis=1, inplace=True)

#Getting rid of strings in DAILY_STRESS column
median_stress = pd.to_numeric(data['DAILY_STRESS'], errors='coerce').median()
data['DAILY_STRESS'] = pd.to_numeric(data['DAILY_STRESS'], errors='coerce').fillna(median_st

#Work life balance - scaled to 0-10 instead of the assumed 0-1000 scale
data["WORK_LIFE_BALANCE_SCORE"] = data["WORK_LIFE_BALANCE_SCORE"]/100
```

```python
data_clean = data.copy()

# One hot encoding AGE and GENDER
encoder = OneHotEncoder(sparse_output=False)
encoded_features = encoder.fit_transform(data_clean[['AGE', 'GENDER']])
encoded_feature_names = encoder.get_feature_names_out(['AGE', 'GENDER'])
# Replace 'AGE' and 'GENDER' with one hot encoded variables
data_clean = data_clean.drop(['AGE', 'GENDER'], axis=1)
data_clean = pd.concat([data_clean, pd.DataFrame(encoded_features, columns=encoded_feature_n

# Split data into training, validation, and test sets
data_train_and_val, data_test = train_test_split(data_clean, test_size=0.2, random_state=11)
data_train, data_val = train_test_split(data_train_and_val, test_size=0.25, random_state=11)


scaler = StandardScaler()
# Apply the scaler to your data and convert to DataFrame instead of array
numerical_features = [col for col in data_train.columns if col not in encoded_feature_names]
data_train[numerical_features] = scaler.fit_transform(data_train[numerical_features])
data_val[numerical_features] = scaler.transform(data_val[numerical_features])
data_test[numerical_features] = scaler.transform(data_test[numerical_features])

# Make csv for future import
data_clean.to_csv('data_clean.csv', index=False)
data_train.to_csv('data_train.csv', index=False)
data_val.to_csv('data_val.csv', index=False)
data_test.to_csv('data_test.csv', index=False)
```

```python
data_clean.head()
```

```
   FRUITS_VEGGIES  DAILY_STRESS  PLACES_VISITED  CORE_CIRCLE  \
0               3           2.0               2            5
1               2           3.0               4            3
2               2           3.0               3            4
```

```
3               3           3.0              10            3
4               5           1.0               3            3

   SUPPORTING_OTHERS  SOCIAL_NETWORK  ACHIEVEMENT  DONATION  BMI_RANGE  \
0                  0               5            2         0          1
1                  8              10            5         2          2
2                  4              10            3         2          2
3                 10               7            2         5          2
4                 10               4            2         4          2

   TODO_COMPLETED  ...  PERSONAL_AWARDS  TIME_FOR_PASSION  WEEKLY_MEDITATION  \
0               6  ...                4                 0                  5
1               5  ...                3                 2                  6
2               2  ...                4                 8                  3
3               3  ...                5                 2                  0
4               5  ...                8                 1                  5

   WORK_LIFE_BALANCE_SCORE  AGE_21 to 35  AGE_36 to 50  AGE_51 or more  \
0                    6.095           0.0           1.0             0.0
1                    6.556           0.0           1.0             0.0
2                    6.316           0.0           1.0             0.0
3                    6.227           0.0           0.0             1.0
4                    6.639           0.0           0.0             1.0

   AGE_Less than 20  GENDER_Female  GENDER_Male
0               0.0            1.0          0.0
1               0.0            1.0          0.0
2               0.0            1.0          0.0
3               0.0            1.0          0.0
4               0.0            1.0          0.0

[5 rows x 27 columns]
```