Importing Required Libraries

```
import warnings
warnings.filterwarnings('ignore')
```

```
import numpy as np
import pandas as pd
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

Exploratory Data Analysis

Load and Prepare Data

```
df1 = pd.read_csv("/content/drive/MyDrive/MFT/mental-and-substance-use-as-share-of-disease.csv")
df2 = pd.read_csv("/content/drive/MyDrive/MFT/prevalence-by-mental-and-substance-use-disorder.csv")
```

```
df1.head()
```

	Entity	Code	Year	DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)
0	Afghanistan	AFG	1990	1.696670
1	Afghanistan	AFG	1991	1.734281
2	Afghanistan	AFG	1992	1.791189
3	Afghanistan	AFG	1993	1.776779
4	Afghanistan	AFG	1994	1.712986

```
df2.head(10)
```

	Entity	Code	Year	Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardized (Percent)
0	Afghanistan	AFG	1990	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	AFG	1991	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	AFG	1992	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	AFG	1993	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	AFG	1994	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779
5	Afghanistan	AFG	1995	0.224713	0.716686	0.111129	4.780851	0.428578	5.098495	0.445422
6	Afghanistan	AFG	1996	0.223690	0.716388	0.107786	4.777272	0.426393	5.100580	0.444837
7	Afghanistan	AFG	1997	0.222424	0.716143	0.103931	4.775242	0.423720	5.105474	0.443938
8	Afghanistan	AFG	1998	0.221129	0.716139	0.100343	4.777377	0.422491	5.113707	0.442665
9	Afghanistan	AFG	1999	0.220065	0.716323	0.097946	4.782067	0.421215	5.120480	0.441428

```
data = pd.merge(df1,df2)
data.head(10)
```

	Entity	Code	Year	DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)	Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence Alcohol us disorders Sex: Both Age: Age standardize (Percent
0	Afghanistan	AFG	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.44403
1	Afghanistan	AFG	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.44425
2	Afghanistan	AFG	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.44550
3	Afghanistan	AFG	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.44595
4	Afghanistan	AFG	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.44577
5	Afghanistan	AFG	1995	1.738272	0.224713	0.716686	0.111129	4.780851	0.428578	5.098495	0.44542
6	Afghanistan	AFG	1996	1.778098	0.223690	0.716388	0.107786	4.777272	0.426393	5.100580	0.44483
7	Afghanistan	AFG	1997	1.781815	0.222424	0.716143	0.103931	4.775242	0.423720	5.105474	0.44393
8	Afghanistan	AFG	1998	1.729402	0.221129	0.716139	0.100343	4.777377	0.422491	5.113707	0.44266

Data Cleaning

```
data.isnull().sum()
```

```
Entity                                                                                0
Code                                                                                690
Year                                                                                  0
DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)    0
Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)              0
Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)           0
Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)           0
Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)          0
Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent)         0
Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)       0
Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardized (Percent)      0
dtype: int64
```

```
data.drop('Code',axis=1,inplace=True)
```

```
data.head(10)
```

	Entity	Year	DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)	Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)	Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardized (Percent)
0	Afghanistan	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779
5	Afghanistan	1995	1.738272	0.224713	0.716686	0.111129	4.780851	0.428578	5.098495	0.445422
6	Afghanistan	1996	1.778098	0.223690	0.716388	0.107786	4.777272	0.426393	5.100580	0.444837
7	Afghanistan	1997	1.781815	0.222424	0.716143	0.103931	4.775242	0.423720	5.105474	0.443938
8	Afghanistan	1998	1.729402	0.221129	0.716139	0.100343	4.777377	0.422491	5.113707	0.442665
9	Afghanistan	1999	1.850988	0.220065	0.716323	0.097946	4.782067	0.421215	5.120480	0.441428

```
data.size,data.shape
```

```
(68400, (6840, 10))
```

```
data.set_axis(['Country','Year','Scizophrenia','Bipolar_disorder','Eating_disorder','Anxiety','drug_usage','depression','alcohol','mental_fit
```

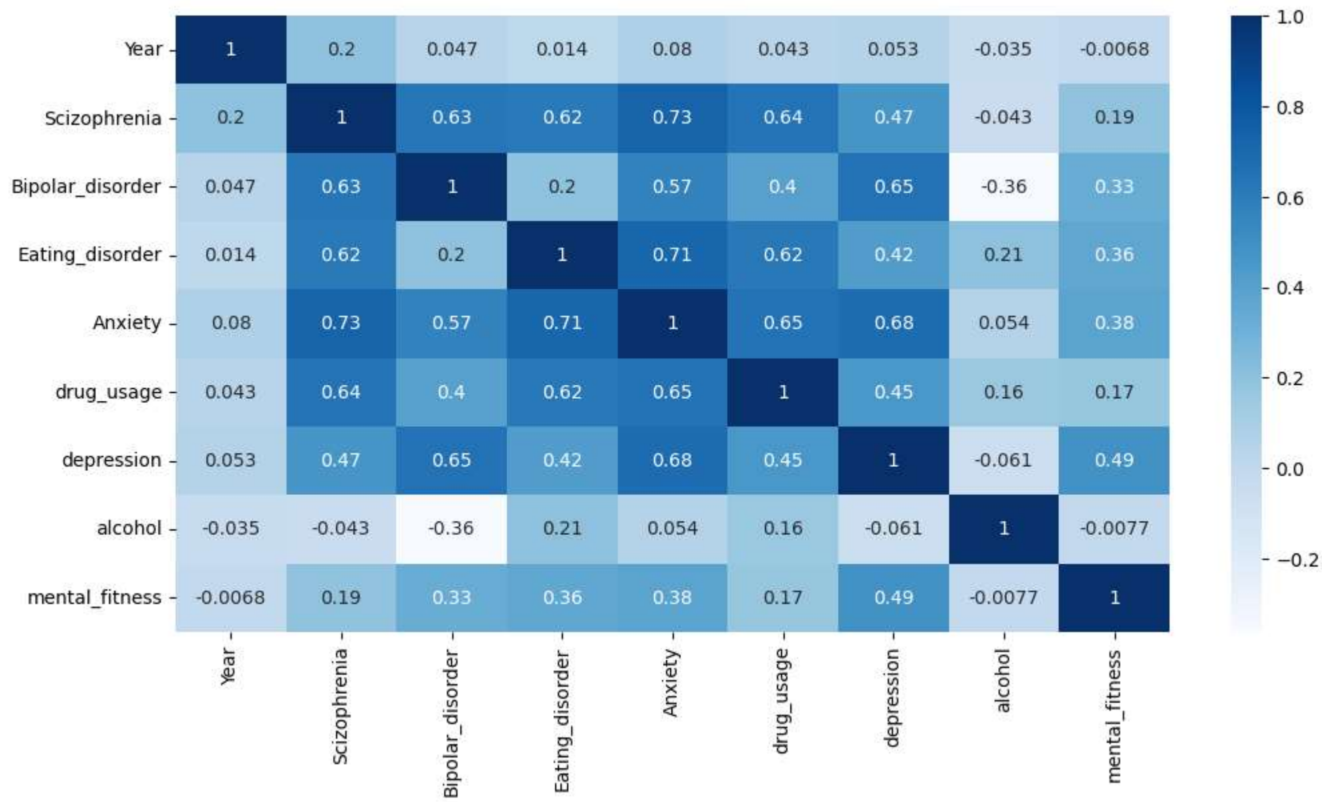
```
data.head(10)
```

	Country	Year	Scizophrenia	Bipolar_disorder	Eating_disorder	Anxiety	drug_usage	depression	alcohol	mental_fitness
0	Afghanistan	1990	1.696670	0.228979	0.721207	0.131001	4.835127	0.454202	5.125291	0.444036
1	Afghanistan	1991	1.734281	0.228120	0.719952	0.126395	4.821765	0.447112	5.116306	0.444250
2	Afghanistan	1992	1.791189	0.227328	0.718418	0.121832	4.801434	0.441190	5.106558	0.445501
3	Afghanistan	1993	1.776779	0.226468	0.717452	0.117942	4.789363	0.435581	5.100328	0.445958
4	Afghanistan	1994	1.712986	0.225567	0.717012	0.114547	4.784923	0.431822	5.099424	0.445779
5	Afghanistan	1995	1.738272	0.224713	0.716686	0.111129	4.780851	0.428578	5.098495	0.445422
6	Afghanistan	1996	1.778098	0.223690	0.716388	0.107786	4.777272	0.426393	5.100580	0.444837
7	Afghanistan	1997	1.781815	0.222424	0.716143	0.103931	4.775242	0.423720	5.105474	0.443938
8	Afghanistan	1998	1.729402	0.221129	0.716139	0.100343	4.777377	0.422491	5.113707	0.442665
9	Afghanistan	1999	1.850988	0.220065	0.716323	0.097946	4.782067	0.421215	5.120480	0.441428

Visualization

```
plt.figure(figsize=(12,6))
sns.heatmap(data.corr(),annot=True,cmap='Blues')
plt.plot()
```

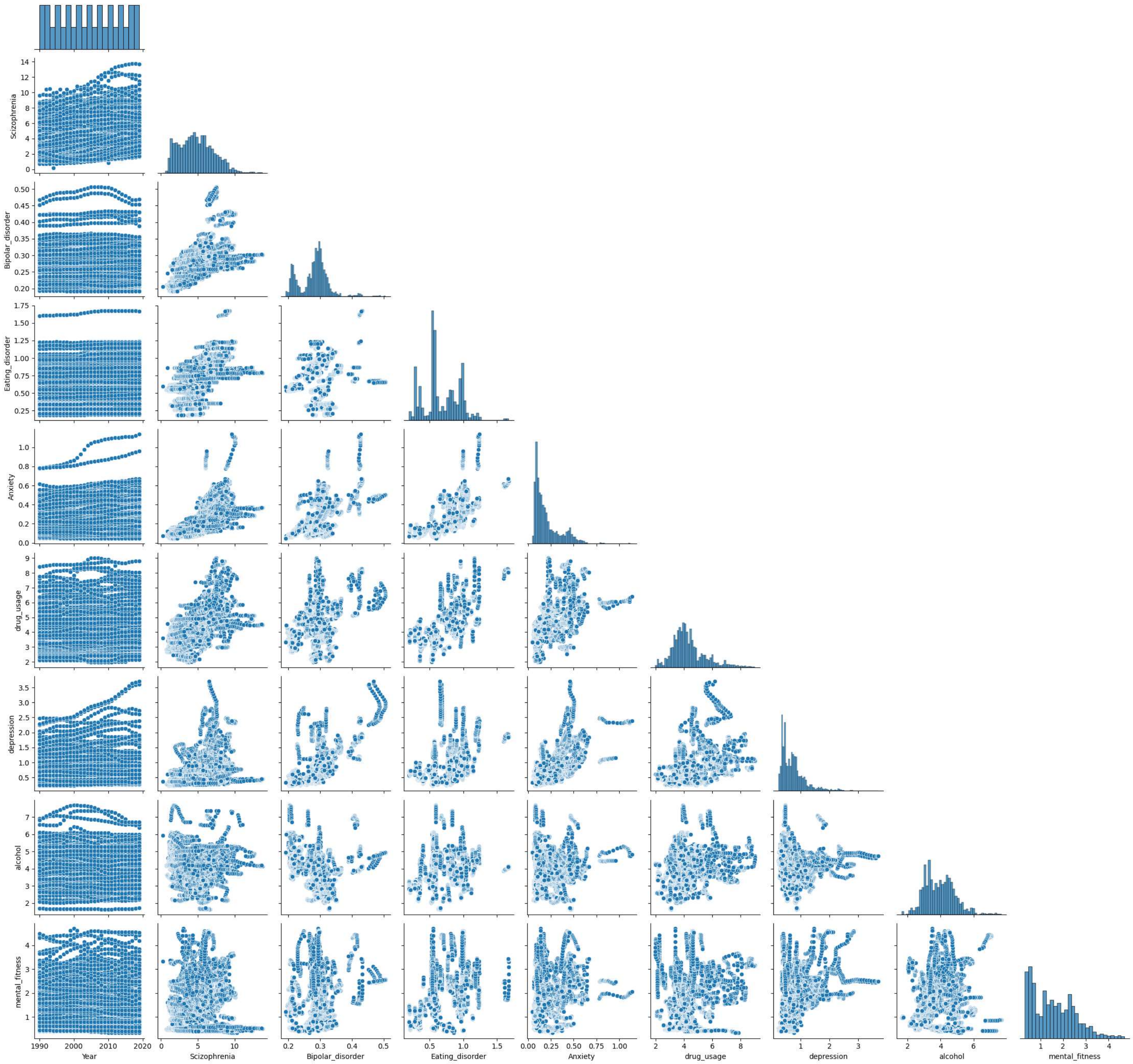
[]



Take Away Points

Eating disorder is a positively correlated to mental fitness and vice versa as our eating choice affect our mental health

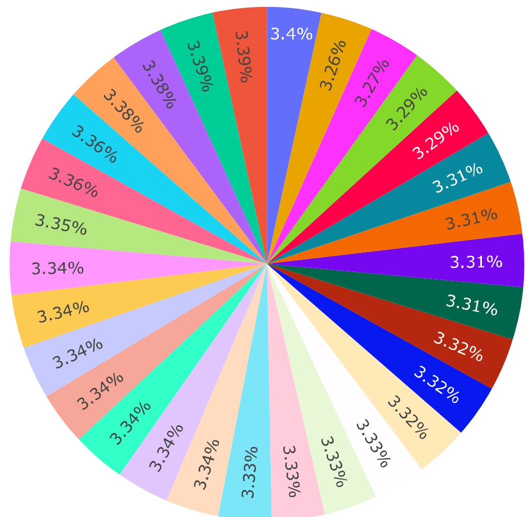
```
sns.pairplot(data,corner=True)
plt.show()
```



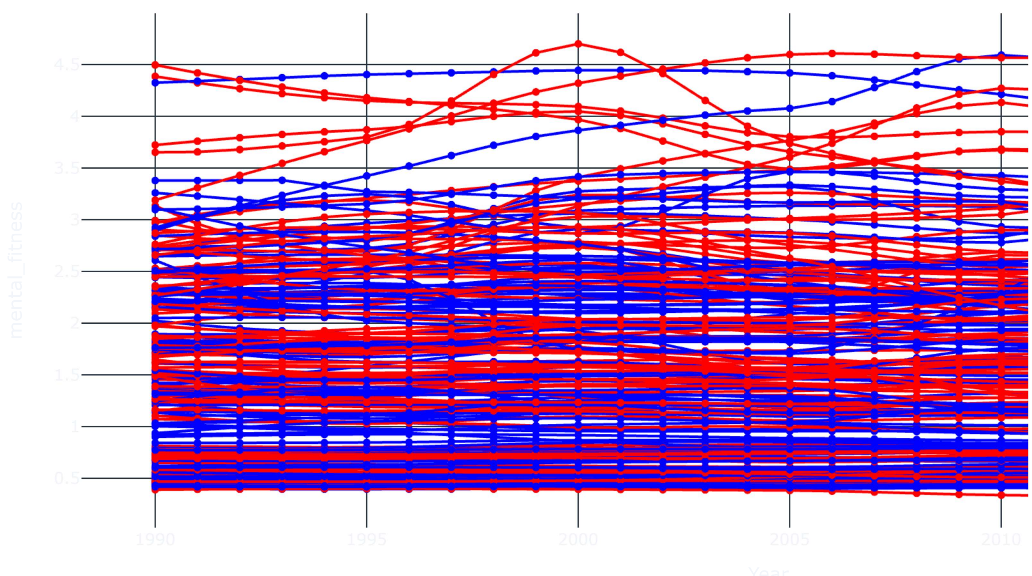

```
mean = data['mental_fitness'].mean()
mean
```

```
1.5788071625382236
```

```
fig = px.pie(data, values='mental_fitness', names='Year')
fig.show()
```




```
fig = px.line(data, x='Year', y='mental_fitness', color ='Country', markers=True, color_discrete_sequence = ['red','blue'], template = 'plotl
fig.show()
```




```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6840 entries, 0 to 6839
Data columns (total 4 columns):
 #   Column                                                                    Non-Null Count  Dtype
---  ------                                                                    --------------  -----
 0   Entity                                                                    6840 non-null   object
 1   Code                                                                      6150 non-null   object
 2   Year                                                                      6840 non-null   int64
```

```
 3   DALYs (Disability-Adjusted Life Years) - Mental disorders - Sex: Both - Age: All Ages (Percent)  6840 non-null   float64
dtypes: float64(1), int64(1), object(2)
memory usage: 213.9+ KB
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6840 entries, 0 to 6839
Data columns (total 10 columns):
 #   Column                                                                        Non-Null Count  Dtype  
---  ------                                                                        --------------  -----  
 0   Entity                                                                        6840 non-null   object 
 1   Code                                                                          6150 non-null   object 
 2   Year                                                                          6840 non-null   int64  
 3   Prevalence - Schizophrenia - Sex: Both - Age: Age-standardized (Percent)      6840 non-null   float64
 4   Prevalence - Bipolar disorder - Sex: Both - Age: Age-standardized (Percent)   6840 non-null   float64
 5   Prevalence - Eating disorders - Sex: Both - Age: Age-standardized (Percent)   6840 non-null   float64
 6   Prevalence - Anxiety disorders - Sex: Both - Age: Age-standardized (Percent)  6840 non-null   float64
 7   Prevalence - Drug use disorders - Sex: Both - Age: Age-standardized (Percent) 6840 non-null   float64
 8   Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)  6840 non-null   float64
 9   Prevalence - Alcohol use disorders - Sex: Both - Age: Age-standardized (Percent)  6840 non-null   float64
dtypes: float64(7), int64(1), object(2)
memory usage: 534.5+ KB
```

```
from sklearn.preprocessing import LabelEncoder
l=LabelEncoder()
for i in df1.columns:
  if df1[i].dtype == 'object':
    df1[i]=l.fit_transform(df1[i])
```

```
from sklearn.preprocessing import LabelEncoder
l=LabelEncoder()
for i in df2.columns:
  if df2[i].dtype == 'object':
    df2[i]=l.fit_transform(df2[i])
```

```
df1.shape
```

```
(6840, 4)
```

```
df2.shape
```

```
(6840, 10)
```

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Define your data and column names
data = {
    'column1': [10, 20, 30, ...],  # Replace with actual values
    'column2': [0.5, 0.3, 0.8, ...],  # Replace with actual values
    'mental_fitness': [1, 0, 1, ...],  # Replace with actual values
    # Add more columns as needed
}

# Create the DataFrame
df = pd.DataFrame(data)

# Continue with your code using the DataFrame 'df'
x = df.drop('mental_fitness', axis=1)
y = df['mental_fitness']
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=2)
```

```
print("xtrain :",xtrain.shape)
print("xtest :",xtest.shape)
print("\nytrain :",ytrain.shape)
print("ytest :",ytest.shape)
```

```
xtrain : (3, 2)
xtest : (1, 2)
```

```
ytrain : (3,)
ytest : (1,)
```

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Example values for xtrain
xtrain = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

# Example values for ytrain
ytrain = np.array([10, 20, 30])

# Reshape ytrain to a 2-dimensional array if needed
ytrain = ytrain.reshape(-1, 1)

lr = LinearRegression()
lr.fit(xtrain, ytrain)
ytrain_pred = lr.predict(xtrain)
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = np.sqrt(mse)
r2 = r2_score(ytrain, ytrain_pred)

print('The Linear Regression model performance for the training set')
print('-------------------------------------------------------')
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

```
The Linear Regression model performance for the training set
-------------------------------------------------------
MSE is 4.2072581611787294e-30
RMSE is 2.0511601988091347e-15
R2 score is 1.0
```

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(xtrain,ytrain)

ytrain_pred = rf.predict(xtrain)
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = np.sqrt(mse)
r2 = r2_score(ytrain, ytrain_pred)

print('The RandomForestRegressor model performance for the training set')
print('-------------------------------------------------------')
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

```
The RandomForestRegressor model performance for the training set
-------------------------------------------------------
MSE is 10.806666666666667
RMSE is 3.287349489583769
R2 score is 0.8379
```

Evaluation

```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
xtrain = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
ytrain = np.array([10, 20, 30])
ytrain = ytrain.reshape(-1, 1)

lr = LinearRegression()
lr.fit(xtrain, ytrain)
ytrain_pred = lr.predict(xtrain)
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = np.sqrt(mse)
r2 = r2_score(ytrain, ytrain_pred)

print('The Linear Regression model performance for the training set')
print('-------------------------------------------------------')
print('MSE is {}'.format(mse))
```

```
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))

from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
rf.fit(xtrain,ytrain)

ytrain_pred = rf.predict(xtrain)
mse = mean_squared_error(ytrain, ytrain_pred)
rmse = np.sqrt(mse)
r2 = r2_score(ytrain, ytrain_pred)

print('\nThe RandomForestRegressor model performance for the training set')
print('-------------------------------------------------------')
print('MSE is {}'.format(mse))
print('RMSE is {}'.format(rmse))
print('R2 score is {}'.format(r2))
```

```
The Linear Regression model performance for the training set
-------------------------------------------------------
MSE is 4.2072581611787294e-30
RMSE is 2.0511601988091347e-15
R2 score is 1.0

The RandomForestRegressor model performance for the training set
-------------------------------------------------------
MSE is 7.183333333333337
RMSE is 2.680174123696693
R2 score is 0.89225
```