**import** pandas **as** pd

**import** numpy **as** np

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**import** os

**for** dirname, \_, filenames **in** os.walk('/kaggle/input'):

**for** filename **in** filenames:

print(os.path.join(dirname, filename))

add Codeadd Markdown

# Loading the data

add Codeadd Markdown

[ ]:



data **=** pd.read\_csv('/kaggle/input/mental-health-in-tech-survey/survey.csv')

add Codeadd Markdown

[ ]:



data.head().T

add Codeadd Markdown

# Data cleaning

add Codeadd Markdown

[ ]:



data.isnull().sum().sort\_values(ascending**=False**)

add Codeadd Markdown

# We can drop columns 'comments', 'state', 'Timestamp' as they are not for our future models

# Column 'state' is only in use if the person lives in USA

add Codeadd Markdown

[ ]:



data.drop('comments',axis**=**1,inplace**=True**)

data.drop('state',axis**=**1,inplace**=True**)

data.drop('Timestamp',axis**=**1,inplace**=True**)

add Codeadd Markdown

[ ]:



data.head().T

add Codeadd Markdown

[ ]:



data.isnull().sum().sort\_values(ascending**=False**)

add Codeadd Markdown

[ ]:



print(data['work\_interfere'].unique())

print(data['self\_employed'].unique())

add Codeadd Markdown

According to data description 'work\_interfere' ask question 'If you have a mental health condition, do you feel that it interferes with your work?'.  
We can assume that people that have left this field empty do not have a diagnosed mental condition. We changed all nan values to 'no answer'.

add Codeadd Markdown

[ ]:



data['work\_interfere'] **=** data['work\_interfere'].fillna('no answer')

print("\n",data['work\_interfere'].unique(),"\n")

print(data.isnull().sum().sort\_values(ascending**=False**))

add Codeadd Markdown

Because there is only 18 data points where we have nan values in column 'self\_employed', we can just drop them

add Codeadd Markdown

[ ]:



data.dropna(inplace**=True**)

​

print("\n",data['self\_employed'].unique(),"\n")

print(data.isnull().sum().sort\_values(ascending**=False**))

add Codeadd Markdown

[ ]:



unique\_values\_per\_column **=** {}

**for** column **in** data.columns:

unique\_values\_per\_column[column] **=** data[column].unique()

add Codeadd Markdown

[ ]:



unique\_values\_per\_column['Age']

add Codeadd Markdown

Because there are only 8 data points where age is not in the livable range (given that the oldest person that ever lived was 122 when she died)  
or negative, or too young to work, we can drop theese

add Codeadd Markdown

[ ]:



uncleaned\_age\_data **=** data[(data['Age'] **<** 18) **|** (data['Age'] **>** 123)]

len(uncleaned\_age\_data)

uncleaned\_age\_data['Age'].index

add Codeadd Markdown

[ ]:



data **=** data.drop(uncleaned\_age\_data['Age'].index)

add Codeadd Markdown

[ ]:



data['Age'].unique()

add Codeadd Markdown

# Feature engineering

add Codeadd Markdown

# Categorising gender

add Codeadd Markdown

[ ]:



unique\_values\_per\_column['Gender']

add Codeadd Markdown

[ ]:



male **=** ['male', 'Male','M', 'm', 'Male-ish', 'maile','Cis Male','Mal', 'Male (CIS)','Make','Male ', 'Man',

'msle','cis male', 'Cis Man','Malr','Mail']

female **=** ['Female', 'female','Cis Female', 'F','f','Femake', 'woman','Female ','cis-female/femme','Female (cis)','femail','Woman','female']

trans **=** ['Trans-female','something kinda male?','queer/she/they','non-binary','All','fluid', 'Genderqueer','Androgyne', 'Agender','Guy (-ish) ^\_^',

'male leaning androgynous','Trans woman','Neuter', 'Female (trans)','queer','ostensibly male, unsure what that really means','trans']

​

data['Gender'].replace(to\_replace**=**male,value**=**'male',inplace**=True**)

data['Gender'].replace(to\_replace**=**female,value**=**'female',inplace**=True**)

data['Gender'].replace(to\_replace**=**trans,value**=**'trans',inplace**=True**)

​

print(data['Gender'].unique())

add Codeadd Markdown

[ ]:



data.drop(data[data['Gender'] **==** 'Nah'].index,inplace**=True**)

data.drop(data[data['Gender'] **==** 'Enby'].index,inplace**=True**)

data.drop(data[data['Gender'] **==** 'A little about you'].index,inplace**=True**)

​

data['Gender'].unique()

add Codeadd Markdown

# Categorising country region

add Codeadd Markdown

[ ]:



unique\_values\_per\_column['Country']

add Codeadd Markdown

# Even though Russia is predominantly in Asia, majority of population lives in Europe

add Codeadd Markdown

[ ]:



north\_america **=** ['United States','Canada','Mexico','Costa Rica','Bahamas, The']

south\_america **=** ['Brazil','Colombia','Uruguay']

europe **=** ['France','United Kingdom','Portugal','Netherlands', 'Switzerland', 'Poland','Germany',

'Russia','Slovenia','Austria','Ireland','Italy', 'Bulgaria', 'Sweden','Latvia', 'Romania',

'Belgium','Spain', 'Finland','Bosnia and Herzegovina', 'Hungary','Croatia', 'Norway','Denmark',

'Greece', 'Moldova','Czech Republic']

asia **=** ['Georgia','Singapore', 'Japan','Thailand','China','Philippines','India','Israel']

africa **=** ['Zimbabwe','Nigeria','South Africa']

oceania **=** ['Australia','New Zealand']

​

data['Country'].replace(to\_replace**=**north\_america, value**=**'NA',inplace**=True**)

data['Country'].replace(to\_replace**=**south\_america, value**=**'SA',inplace**=True**)

data['Country'].replace(to\_replace**=**europe, value**=**'EU',inplace**=True**)

data['Country'].replace(to\_replace**=**asia, value**=**'AS',inplace**=True**)

data['Country'].replace(to\_replace**=**africa, value**=**'AF',inplace**=True**)

data['Country'].replace(to\_replace**=**oceania, value**=**'OC',inplace**=True**)

​

data['Country'].unique()

​

​

add Codeadd Markdown

# Simple EDA

add Codeadd Markdown

[ ]:



data.info()

add Codeadd Markdown

[ ]:



sns.set(rc**=**{'figure.figsize':(20,10)})

sns.set\_theme(style**=**"darkgrid")

sns.countplot(x**=**'treatment',data**=**data)

​

add Codeadd Markdown

[ ]:



sns.countplot(x**=**'treatment',hue**=**'Gender',data**=**data)

add Codeadd Markdown

[ ]:



sns.countplot(x**=**'treatment',hue**=**'family\_history',data**=**data)

add Codeadd Markdown

[ ]:



sns.countplot(x**=**'family\_history',hue**=**'Gender',data**=**data)

add Codeadd Markdown

[ ]:



sns.countplot(x**=**'Country',data**=**data,order**=**data['Country'].value\_counts().index)

add Codeadd Markdown

[ ]:



sns.countplot(x**=**'treatment',hue**=**'Country',data**=**data)

add Codeadd Markdown

[ ]:



sns.countplot(x**=**'no\_employees',data**=**data)

add Codeadd Markdown

[ ]:



sns.countplot(x**=**'treatment',hue**=**'no\_employees',data**=**data)

add Codeadd Markdown

[ ]:



sns.countplot(y**=**'Age',data**=**data)

add Codeadd Markdown

[ ]:



sns.boxplot(y**=**'Age',data**=**data)

add Codeadd Markdown

# To be continued...

# Things that will be added: Spliting data into folds, hyperparameter optimization for our models of choice and model result comparrison

add Codeadd Markdown