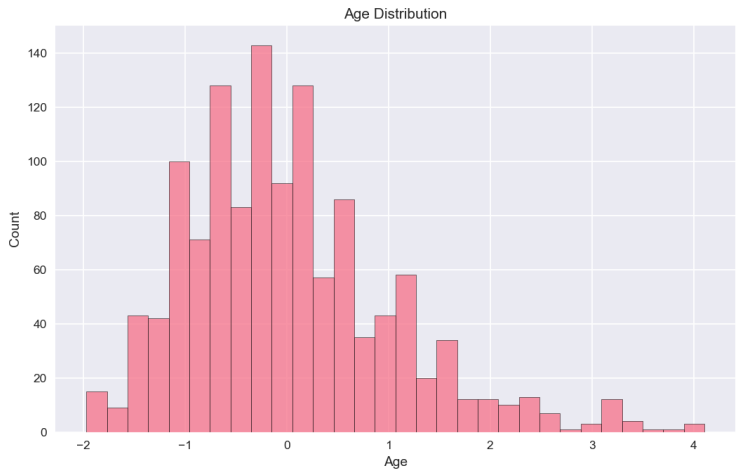
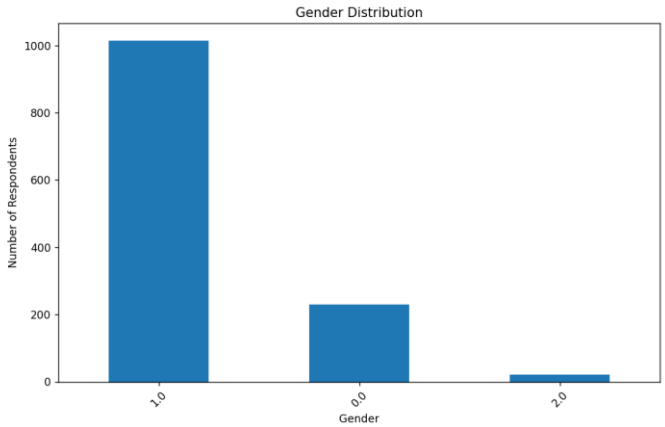
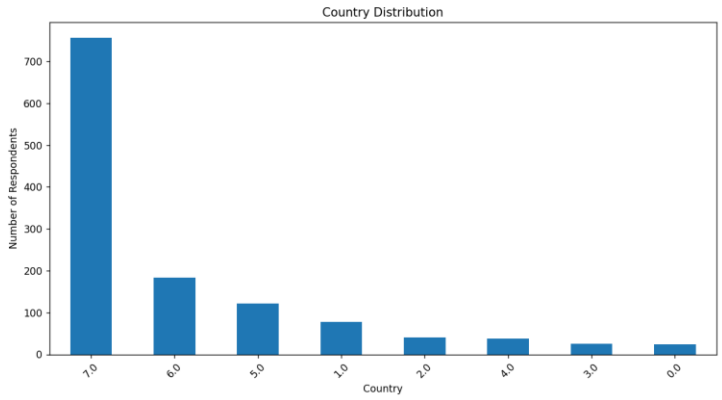
Data Collection and Preprocessing Phase

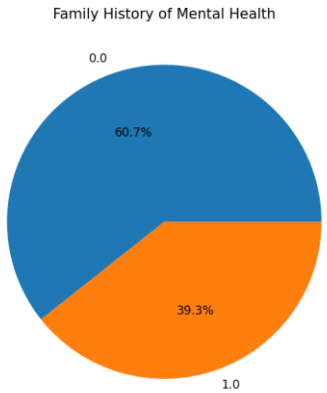
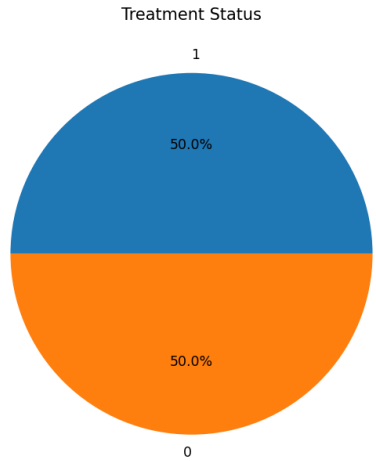
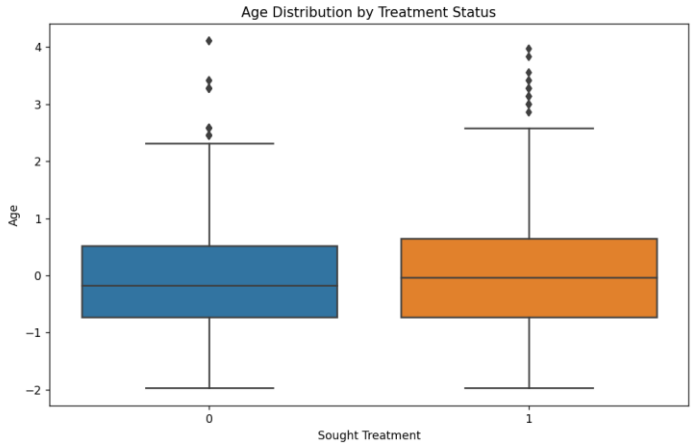
Date	16 June 2025
Team ID	SWTID1749709635
Project Title	Mental Health Prediction
Maximum Marks	6 Marks

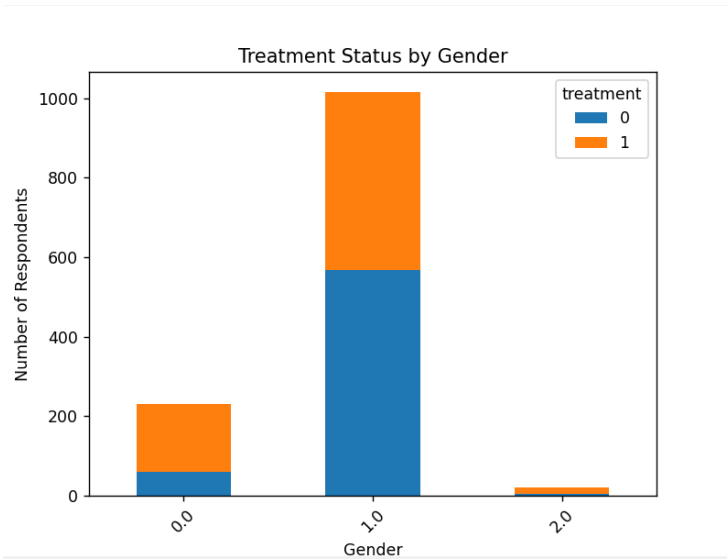
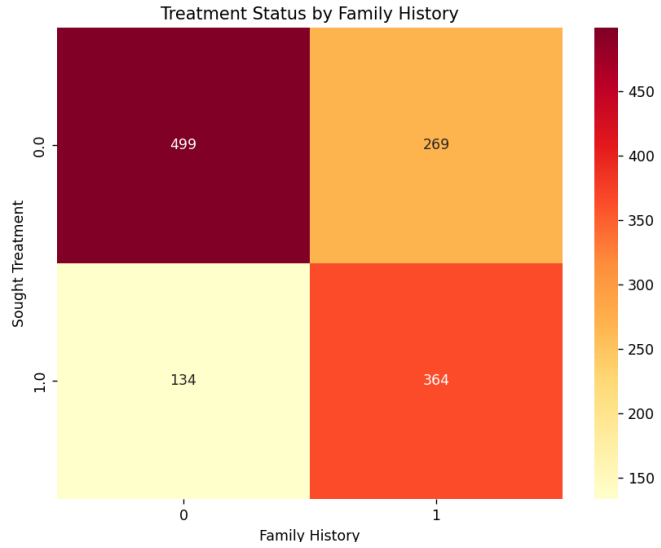
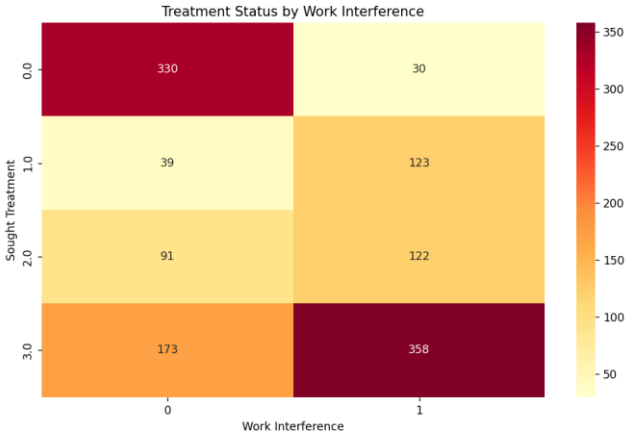
Data Exploration and Preprocessing Template

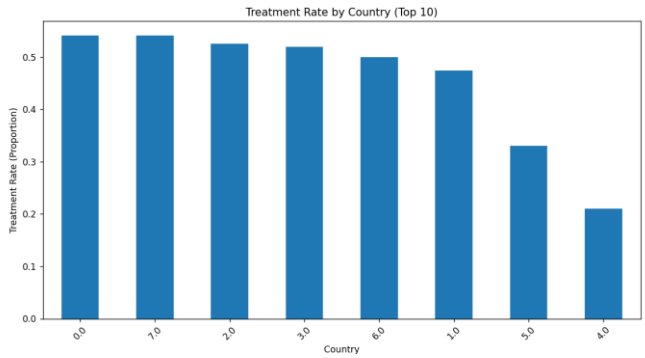
Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks such as normalization and feature engineering. Data cleaning will address missing values, duplicates, and inconsistencies, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for actionable insights and accurate mental health risk predictions.

Section	**Description**
Data Overview	<u>Dimension:</u> 1260 rows x 27 columns <u>Descriptive statistics:</u> Age count 1245.000000 mean 32.060241 std 7.352870 min 5.000000 25% 27.000000 50% 31.000000 75% 36.000000 max 72.000000

Univariate Analysis	  

Bivariate Analysis	  

	
Multivariate Analysis	 



Outliers and Anomalies

-

Data Preprocessing Code Screenshots

Loading Data

```
#importing the dataset which is in csv filw
df = pd.read_csv('survey.csv')
df
```

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interfere	no_employees	...	leave	mental_healt
0	2014-08-27 11:29:31	37	Female	United States	IL	NaN	No	Yes	Often	6-25	...	Somewhat easy	
1	2014-08-27 11:29:37	44	M	United States	IN	NaN	No	No	Rarely	More than 1000	...	Don't know	
2	2014-08-27 11:29:44	32	Male	Canada	NaN	NaN	No	No	Rarely	6-25	...	Somewhat difficult	
3	2014-08-27 11:29:46	31	Male	United Kingdom	NaN	NaN	Yes	Yes	Often	26-100	...	Somewhat difficult	
4	2014-08-27 11:30:22	31	Male	United States	TX	NaN	No	No	Never	100-500	...	Don't know	

Handling Missing Data

```
# Drop irrelevant columns
df_processed= df_processed.drop(columns=['Timestamp', 'comments'])

# Handle missing values
df_processed['state'] = df_processed['state'].fillna('Unknown')
df_processed['self_employed'] = df_processed['self_employed'].fillna('No')
df_processed['work_interfere'] = df_processed['work_interfere'].fillna('Never')

df_processed = df_processed[(df_processed['Age'] >= 0) & (df_processed['Age'] <= 100)]
df_processed['Gender'] = df_processed['Gender'].str.strip()  # Remove any leading or trailing spaces
```

Data Transformation	
	``python
	\# Standardize entries
	gender_mapping = {
	'M': 'Male', 'male': 'Male', 'Male-ish': 'Male', 'maile': 'Male',
	'Trans-female': 'Trans', 'Cis Female': 'Female', 'F': 'Female',
	'something kinda male?': 'Other', 'Cis Male': 'Male', 'Woman': 'Female',
	'f': 'Female', 'Mal': 'Male', 'Male (CIS)': 'Male', 'queer/she/they': 'Non-binary',
	'non-binary': 'Non-binary', 'Femake': 'Female', 'woman': 'Female', 'Make': 'Male',
	'Nah': 'Other', 'All': 'Other', 'Enby': 'Other', 'fluid': 'Other',
	'Genderqueer': 'Non-binary', 'Female ': 'Female', 'Androgyne': 'Other',
	'Agender': 'Non-binary', 'cis-female/femme': 'Female', 'Guy (-ish) ^_^': 'Other',
	'male leaning androgynous': 'Male', 'Male ': 'Male', 'Man': 'Male',
	'Trans woman': 'Trans', 'msle': 'Male', 'Neuter': 'Other', 'Female (trans)': 'Trans',
	'queer': 'Non-binary', 'Female (cis)': 'Female', 'Mail': 'Male', 'cis male': 'Male',
	'A little about you': 'Other', 'Malr': 'Male', 'p': 'Other', 'femail': 'Female',
	'Cis Man': 'Male', 'ostensibly male, unsure what that really means': 'Other',
	'female': 'Female', 'm': 'Male'
	}
	
	\# Apply the mapping
	df_processed['Gender'] = df_processed['Gender'].map(gender_mapping).fillna(df_processed['Gender'])
	
	df_processed = df_processed[df_processed['Gender'] != 'Other']
	
	\# Standardize categorical variables
	df_processed['Gender'] = df_processed['Gender'].str.lower()
	features = ['Age', 'Gender', 'Country', 'self_employed', 'family_history',
	'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']
	
	categorical_columns = ['Gender', 'Country', 'self_employed', 'family_history',
	'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']
	
	\# One-hot encode categorical variables
	df_features = pd.get_dummies(df_processed[features], drop_first=True)
	
	\# Scale Age
	scaler = StandardScaler()
	df_processed['Age'] = scaler.fit_transform(df_processed[['Age']])
	
	\# Encode target
	le_target = LabelEncoder()
	y_encoded = le_target.fit_transform(df_processed['treatment'])
	``

Feature Engineering	Attached the codes in final submission.
Save Processed Data	-