Mental Health Tech Survey data analysis and predictions

Introduction

This report summarises analysis and predictions of the data titled ‘Mental Health Tech Survey’ hosted at kaggle.com

The aim of this activity is to predict if a tech employee is inclined towards seeking treatment for a mental health condition and how certain factors affect his/her decision towards this.

Abstract

As we know, burnout is common in the fast-paced, competitive environment of the tech industry which gives rise to increasing stressful days at work leading to mental health issues. We are interested in gauging how mental health is viewed within the tech/IT workplace, and the prevalence of certain mental health disorders within the tech industry.

From certain attributes such as the treatment of health issues by the correspondent’s coworkers and employers to his/her own understanding of own health issues, we intend to predict if the employees would ever seek professional treatment in reponse to their plight.

This report starts by stating the problem statement upon which the predictions and analysis are based followed by a brief description of the data. Data preprocessing, exploratory data analysis and predictions for the problem statement via various classification models are further steps taken to bring this activity to fulfilment.

Problem Statement

Predict if a certain employee would seek mental health treatment given attributes of some personal details and corresponding work atmosphere.

Overview of Data

The dataset is hosted at https://www.kaggle.com/osmi/mental-health-in-tech-survey/home for 2014 survey with 1259 rows X 27 columns
and at https://www.kaggle.com/osmi/mental-health-in-tech-2016 for 2016 survey with 1433 rows X 27 columns. We use a merged version of these datasets.

This dataset is from 2014 and 2016 surveys that measures attitudes towards mental health and frequency of mental health disorders in the tech workplace.

This dataset contains the following data:

- **Timestamp**
- **Age**
- **Gender**
- **Country**
- **state**: If you live in the United States, which state or territory do you live in?
- **self_employed**: Are you self-employed?
- **family_history**: Do you have a family history of mental illness?
- **treatment**: Have you sought treatment for a mental health condition?
- **work_interfere**: If you have a mental health condition, do you feel that it interferes with your work?
- **no_employees**: How many employees does your company or organization have?
- **remote_work**: Do you work remotely (outside of an office) at least 50% of the time?
- **tech_company**: Is your employer primarily a tech company/organization?
- **benefits**: Does your employer provide mental health benefits?
- **care_options**: Do you know the options for mental health care your employer provides?
- **wellness_program**: Has your employer ever discussed mental health as part of an employee wellness program?
- **seek_help**: Does your employer provide resources to learn more about mental health issues and how to seek help?
- **anonymity**: Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources?
- **leave**: How easy is it for you to take medical leave for a mental health condition?

- **mental_health_consequence**: Do you think that discussing a mental health issue with your employer would have negative consequences?
- **phys_health_consequence**: Do you think that discussing a physical health issue with your employer would have negative consequences?
- **coworkers**: Would you be willing to discuss a mental health issue with your coworkers?
- **supervisor**: Would you be willing to discuss a mental health issue with your direct supervisor(s)?
- **mental_health_interview**: Would you bring up a mental health issue with a potential employer in an interview?
- **phys_health_interview**: Would you bring up a physical health issue with a potential employer in an interview?
- **mental_vs_physical**: Do you feel that your employer takes mental health as seriously as physical health?
- **obs_consequence**: Have you heard of or observed negative consequences for coworkers with mental health conditions in your workplace?
- **comments**: Any additional notes or comments

Most of the data is categorical in nature which can be encoded to be used directly in our models. There are a few missing values in some columns and some inconsistencies in attributes such as Gender. Since the data was collected via a survey, these inconsistencies are to be expected.

Data Preprocessing

We start by identifying null values and replacing them appropriately.

First column with null values is **self_employed.**

```
In [4]: df.self_employed.isnull().sum()
Out[4]: 18
```

Checking counts of values.

```
In [5]: df.self_employed.value_counts()
Out[5]: No     1095
        Yes     146
        Name: self_employed, dtype: int64
```

Since we have ‘No’ way more than ‘Yes’, we can replace 18 NaNs with ‘No’.

```
In [6]: df.self_employed.fillna(value="No", inplace=True)
```

Similarly, for **work_interfere** we have 64 null values and the label ‘Sometimes’ more than others. So we replaces NaNs with it.

```
In [7]: df.work_interfere.value_counts()
Out[7]: Sometimes    465
        Never        213
        Rarely       173
        Often        144
        Name: work_interfere, dtype: int64

In [8]: df.work_interfere.fillna(value="Sometimes", inplace=True)
```

In the field **Gender** we got different labels corresponding to the same meaning, such as Male is labelled as ‘m’, ‘M’, ‘male’, ‘Male’ including grammatically incorrect labels such as ‘mail’, ‘make’ etc. Similar is with Female and Trans that is labelled differently. Another inconsistency that arose due to the method of surveying.
To correct this, we identify all what we can and finally label everything as ‘male’, ‘female’ and ‘trans’.

```
male_str = ["male","male.","male/genderqueer", "m","cisdude", "male-ish", "maile", "mal", "male (cis)", "make", "male ", "ma
 'male 9:1 female, roughly',"i'm a man why didn't you make this a drop down question. you should of asked sex? and i would
trans_str = ["mtf","other","genderfluid","human",'none of your business',"afab","fm","male (trans, ftm)","other/transfemini
 'genderflux demi-girl' ,'unicorn',"nonbinary","bigender","trans-female", "something kinda male?", "nb masculine","queer/sh
female_str = ["female or multi-gender femme", "female assigned at birth ","female-bodied; no feelings about gender","i iden
              "female (cis)", "femail","cisgender Female","cis female","cis female ","female/woman", "cisgender female"]

for (row, col) in df.iterrows():

    if str.lower(col.Gender) in male_str:
        df['Gender'].replace(to_replace=col.Gender, value='male', inplace=True)

    if str.lower(col.Gender) in female_str:
        df['Gender'].replace(to_replace=col.Gender, value='female', inplace=True)

    if str.lower(col.Gender) in trans_str:
        df['Gender'].replace(to_replace=col.Gender, value='trans', inplace=True)
```

Finally we get values in **df.Gender** as [‘male’, ‘female’, ‘trans’].

Values in **Age** range from negative to 99999, but the actual values can be between 18 and 100. So we replace values other than those in this range with NaN and then replace NaN values in **Age** with median of given values.

```
df['Age'].fillna(df['Age'].median(), inplace = True)
```

Similarly in other columns, we replace terms such as ‘I dont know’, ‘I am not sure’ etc with a single term ‘Don’t know’.

```
df.obs_consequence.fillna(value="No",inplace=True)
df.anonymity.fillna(value="Don't know",inplace=True)
df.Gender.fillna(value="Male",inplace=True)
df.loc[df.anonymity=="I don't know",'anonymity']="Don't know"
df.loc[df.benefits=="Not eligible for coverage / N/A",'benefits']="No"
df.loc[df.benefits=="I don't know",'benefits']="Don't know"
df.benefits.fillna(value="No",inplace=True)
df.loc[df.care_options=="I am not sure",'care_options']="Not sure"
df.care_options.fillna(value="No",inplace=True)
df.coworkers.fillna(value="No",inplace=True)
df.leave.fillna(value="I don't know",inplace=True)
df.mental_health_consequence.fillna(value="No",inplace=True)
df.loc[df.mental_vs_physical=="I don't know",'mental_vs_physical']="Don't know"
df.mental_vs_physical.fillna(value="Don't know",inplace=True)
df.no_employees.fillna(value="26-100",inplace=True)
df.phys_health_consequence.fillna(value="No",inplace=True)
df.loc[df.seek_help=="I don't know",'seek_help']="Don't know"
df.seek_help.fillna(value="No",inplace=True)
df.supervisor.fillna(value="No",inplace=True)
df.loc[df.tech_company==1.0,'tech_company']="Yes"
df.loc[df.tech_company==0.0,'tech_company']="Yes"
df.tech_company.fillna(value="No",inplace=True)
df.loc[df.wellness_program=="I don't know",'wellness_program']="Don't know"
df.wellness_program.fillna(value="No",inplace=True)
```

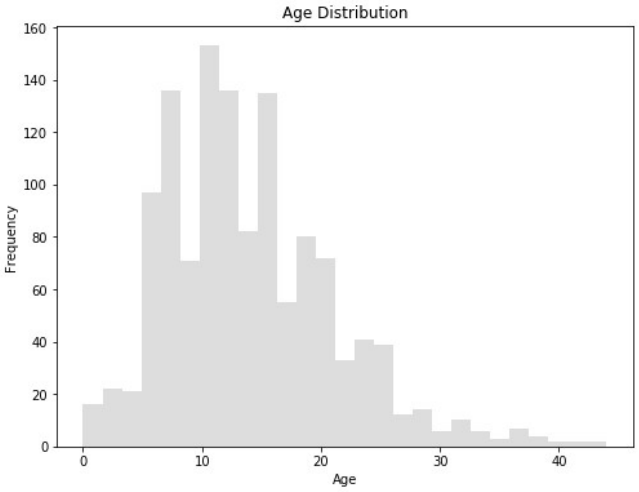
Finally missing values and inconsistencies are done with, so we do label encoding for the remaining categorical data and move on to data exploration.

Data Exploration

Checking the age distribution if it is skewed for a particular age.

```
fig,ax = plt.subplots(figsize=(8,6))
sns.distplot(df['Age'].dropna(),ax=ax,kde=False,color='#aaaaaa')
plt.title('Age Distribution')
plt.ylabel('Frequency')
```

```
Text(0,0.5,'Frequency')
```

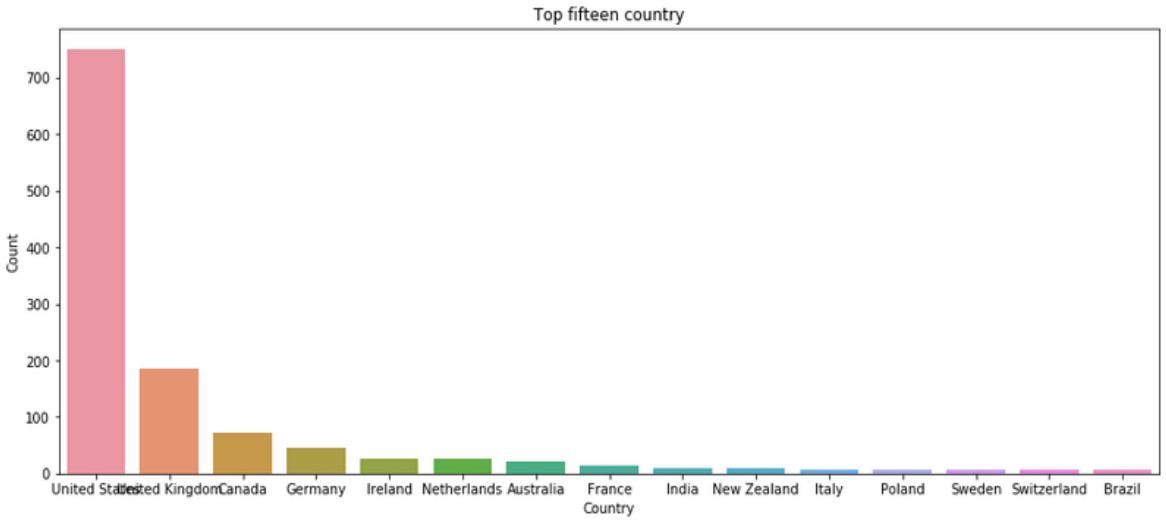


This forms a right skewed bell curve which should be just fine being used in models.

Checking geospatial distribution of data in form of countries.

```
country_count = Counter(df['Country'].tolist()).most_common(15)
country_idx = [country[0] for country in country_count]
country_val = [country[1] for country in country_count]
plt.subplots(figsize=(15,6))
sns.barplot(x = country_idx,y=country_val )
plt.title('Top fifteen country')
plt.xlabel('Country')
plt.ylabel('Count')
```

```
Text(0,0.5,'Count')
```

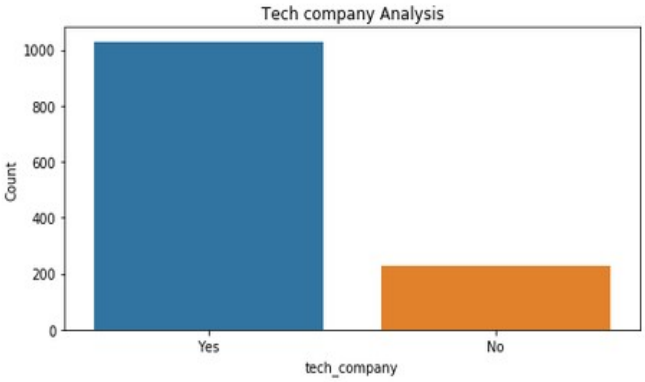


Significant amount of data is from the US followed by UK and Canada. Rest others form a very small portion of the data. So we can ignore regional dependencies.

From below plot we see most of data is from people working in tech.

```
plt.subplots(figsize=(8,4))
sns.countplot(df['tech_company'].dropna())
plt.title('Tech company Analysis')
plt.ylabel('Count')
```

```
Text(0,0.5,'Count')
```

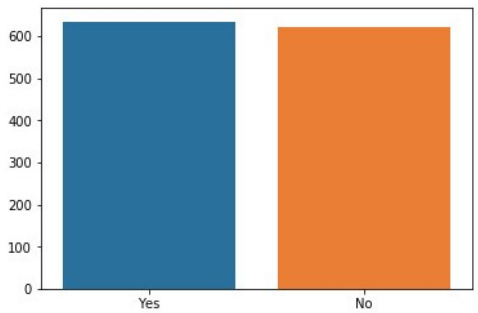


Thus findings can be generalised to tech industry.

Lets see the distribution of labels that need to be predicted.

```
int_level = df['treatment'].value_counts()
sns.barplot(int_level.index, int_level.values)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7e57bb8490>
```



We can see that there is an equally likely chance of a person opting for treatment or not as both values are almost same with ‘Yes’ on a slightly higher side. In other words, data is very balanced.

Models used and corresponding predictions

We split the complete dataframe into two, keeping ‘test.csv’ for further testing and using the remaining data for building our model.

We split remaining data into test and train in ratio of 20:80 for building and validating our model.

We use corresponding features for our predictions:
'Age', 'Gender', 'family_history', 'benefits', 'care_options', 'anonymity', 'leave', 'work_interfere'

Our label to be predicted is ‘treatment’

Accuracy on validation test data using:

1. Decision Tree Classifier : 0.6715481171548117
2. Logistic Regression : 0.5774058577405857
3. K-nearest Neighbour: 0.6631799163179917
4. Random Forest : 0.7217573221757322

Thus, the best result is from using Random Forest Classifier.