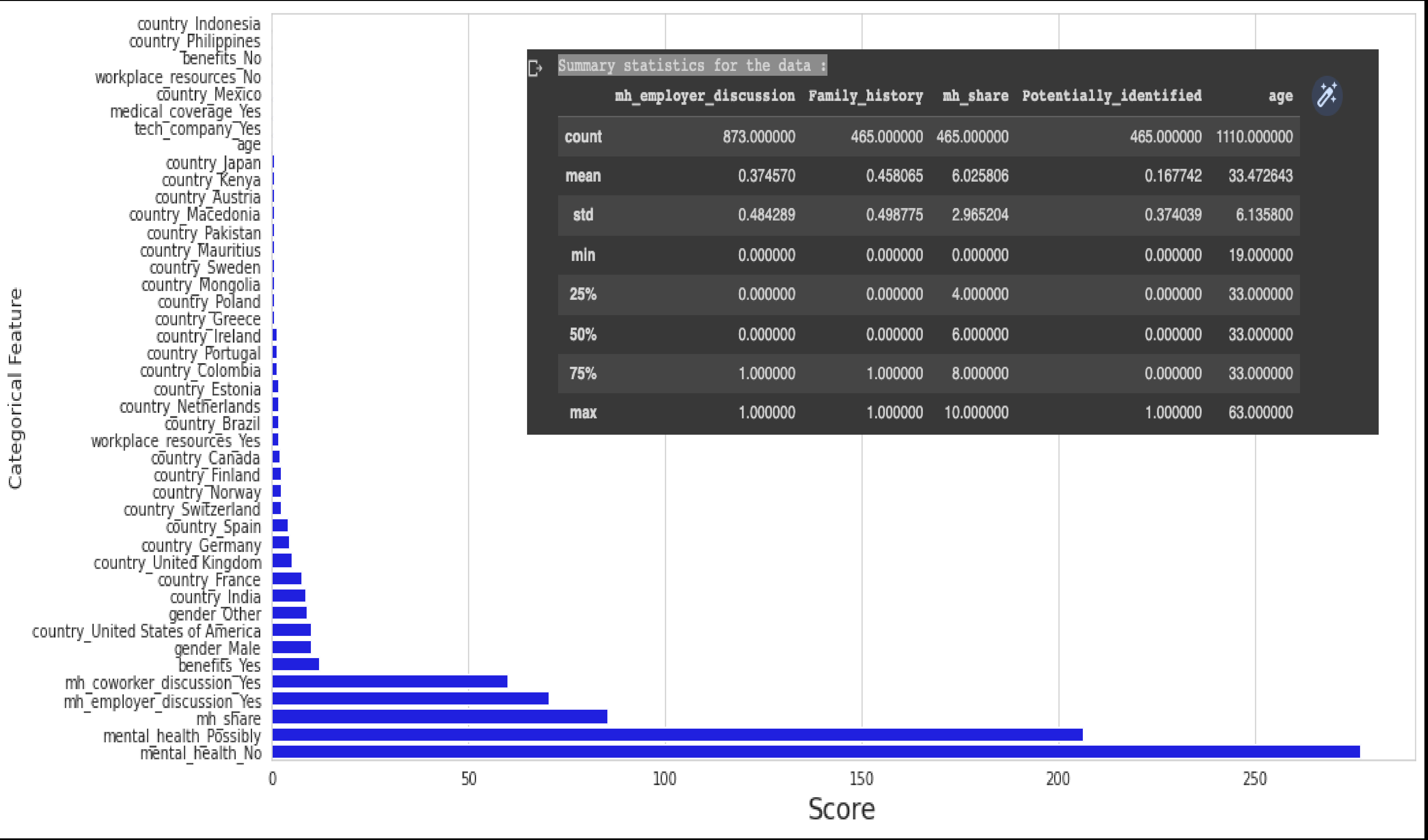
DATA 606: Capstone In Data Science (Phase:2)

By- Adishree Pandey and Shekha Desai (Initiators)

Data Exploration: Data Cleaning of rows and columns

1. Reading the dataset and renaming the columns
2. Removing unnecessary columns
3. Checking and removing the empty rows
4. Removing the duplicates
5. Calculating every unique value in each column and input missing values
6. Removing the Noise from Age & Tokenization in Gender and Benefits Column
7. Countries with 100% Medical Coverage
8. Sorting the data as '1=Yes' or '0=No' according to Countries with Medical Coverage
9. Filtering the Not Sure rows and setting up the TRUE/FALSE rows as 1&0 for all relevant columns
10. Addressing the irregularities in Age with the median
11. Feature Reduction
12. Age Discretization



Summary statistics for the data :

	mh_employer_discussion	Family_history	mh_share	Potentially_identified	age
count	873.000000	465.000000	465.000000	465.000000	1110.000000
mean	0.374570	0.458065	6.025806	0.167742	33.472643
std	0.484289	0.498775	2.965204	0.374039	6.135800
min	0.000000	0.000000	0.000000	0.000000	19.000000
25%	0.000000	0.000000	4.000000	0.000000	33.000000
50%	0.000000	0.000000	6.000000	0.000000	33.000000
75%	1.000000	1.000000	8.000000	0.000000	33.000000
max	1.000000	1.000000	10.000000	1.000000	63.000000

Snippets

```
# lets check number of empty rows in data
data_df.isna().sum().sort_values()
```

```
self_employed               15
mental_health               15
tech_company               285
tech_related_role          285
benefits                   285
workplace_resources        285
mh_employer_discussion     285
mh_coworker_discussion     297
mh_share                  1071
age                       1071
country                   1071
gender                    1083
medical_coverage          1734
dtype: int64
```

```
Other            933
Male             483
Female           300
cis-het male       3
Name: gender, dtype: int64
```

```
[ ] data_df.columns
```

```
Index(['self_employed', 'tech_company', 'tech_related_role', 'benefits',
       'workplace_resources', 'mh_employer_discussion',
       'mh_coworker_discussion', 'medical_coverage', 'mental_health',
       'mh_share', 'age', 'gender', 'country'],
      dtype='object')
```

```
# So, all of the records are missing :/
# lets try to fill this column

# If a company is providing them health benefits, that means they have a medical coverage.
data_df.loc[data_df['benefits'] == 'Yes', 'medical_coverage'] = 'Yes'

# Accoridng to law from UK, all employees are covered for medical health, so lets update all residents who reside in UK.
data_df.loc[data_df['country'] == 'United Kingdom', 'medical_coverage'] = 'Yes'

# According to OECD-ilibrary.org, following countries have 100% record for individuals with health benefits, so lets update those as well.
countries = ['Germany', 'Canada', 'France', 'Spain', 'Netherlands']
data_df.loc[(data_df['country'].isin(countries)), 'medical_coverage'] = 'Yes'

# Lets check how many null values we have now
data_df['medical_coverage'].isna().sum()
```

```
795
```

```
other                      918
male                       429
female                     237
f                           42
m                           39
woman                       15
afab non-binary              3
masculine                    3
female (cis)                 3
b                            3
male/he/him                  3
homem cis                    3
cis-het male                 3
cis male                     3
non-binary                   3
non-binary/agender           3
female, she/her              3
mostly male                  3
cisgender male               3
Name: gender, dtype: int64
```

Feature reduction

Lets do some feature reduction including re formating the values.

- self_employed: From 0/1 to No/Yes
- tech_company: From 0.0/1.0 to No/Yes
- seek_help: From 0/1 to No/Yes
- no_employees: 'More than 1000' to '>1000'
- mh_employer_discussion: From 0.0/1.0 to No/Yes
- mh_coworker_discussion: From 0.0/1.0 to No/Yes

```
TRUE     846
1        720
0         87
FALSE     66
Name: tech_related_role, dtype: int64
```

```
count    295.000000
mean      35.494915
std        8.901911
min       19.000000
25%       29.000000
50%       35.000000
75%       41.500000
max       63.000000
Name: age, dtype: float64
```

After Data Cleaning, we conclude that:

Adult=90.8,
Youth=9.2,
Senior=0

More than 60% employee have medical coverage provided from employer, but not the resources to get more information, suggesting that companies do not get active involvement.

Around 12% employee do not have medical coverage.

Overal 67% feel difficult to discuss the MH issues with employer, hence never discussed.

Most Men in their 30s in our data have Mental health issues.

More female employees have mental health disorders than those who don't.

Mostly employees whose gender falls under 'Others' catgory hav less to no mental health issues.

67% of employees are effected with mental illness overall.

Introducing Machine Learning

Decision Tree Classifier

	precision	recall	f1-score	support
0.0	0.69	0.41	0.51	22
1.0	0.67	0.87	0.75	30
accuracy			0.67	52
macro avg	0.68	0.64	0.63	52
weighted avg	0.68	0.67	0.65	52

	precision	recall	f1-score	support
0.0	0.50	0.27	0.35	22
1.0	0.60	0.80	0.69	30
accuracy			0.58	52
macro avg	0.55	0.54	0.52	52
weighted avg	0.56	0.58	0.54	52

Gaussian Naïve Bayes Classifier

Training set
score: 0.7476

Model accuracy
score:0.6154

Test set score:
0.6154

Training-set accuracy
score: 0.7476

SVC

Accuracy: 0.576923
0769230769

Random Forest Classifier

Accuracy: 0.673076
9230769231

	precision	recall	f1-score	support
0.0	0.69	0.41	0.51	22
1.0	0.67	0.87	0.75	30
accuracy			0.67	52
macro avg	0.68	0.64	0.63	52
weighted avg	0.68	0.67	0.65	52

	precision	recall	f1-score	support
0.0	0.50	0.27	0.35	22
1.0	0.60	0.80	0.69	30
accuracy			0.58	52
macro avg	0.55	0.54	0.52	52
weighted avg	0.56	0.58	0.54	52

Logistic Regression

	precision	recall	f1-score	support
0.0	0.55	0.27	0.36	22
1.0	0.61	0.83	0.70	30
accuracy			0.60	52
macro avg	0.58	0.55	0.53	52
weighted avg	0.58	0.60	0.56	52

Hyperparameter Tuning

Recall Score is
0.8333333333333334

Testing set Score
Is 88.80363739698778

ROC Score
is 0.7121212121212121

Training set
score: 97.09%

Thank You