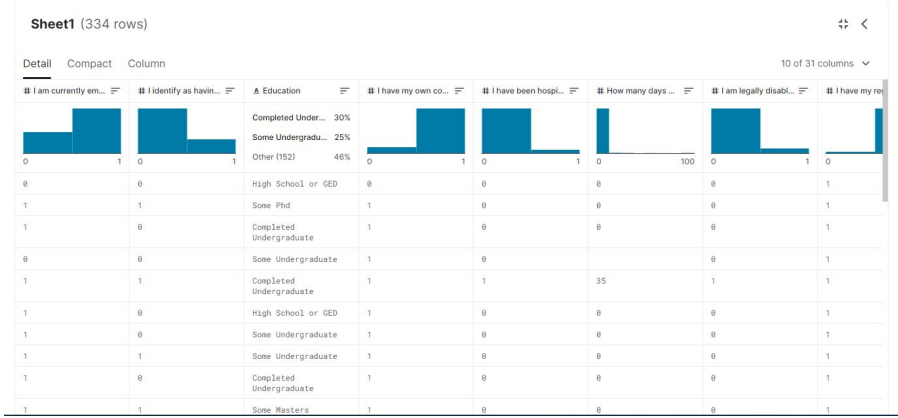
Relationship between Mental Illness and Unemployment

By Jerome Veix and Williem Christian Encarnacion

Background surrounding the data

- Where to find dataset: https://www.kaggle.com/datasets/michaelacorley/unemployment-and-mental-illness-survey
- Context of data: The dataset explores the relationship between mental illness and high unemployment rate. This data was collected by the National Alliance of Mental Illness (NAMI) through surveys. Only 20-25% of the participants are mentally ill representing the actual ratio within society.
- What we're looking at: We will be analyzing behavioral and societal factors in their linkage with employment.
- Type of question: Classification problem



EDA I

- Most of the data is categorical variables with binary outputs(0 or 1)
- Describe function only applies to 25 out of the 31 columns
- Further proves why we should use classification models as there aren't many columns with continuous values

```
df.describe()
```

	I am currently employed at least part-time	I identify as having a mental illness	I have my own computer separate from a smart phone	I have been hospitalized before for my mental illness	How many days were you hospitalized for your mental illness	I am legally disabled	I have my regular access to the internet	I live with my parents	I have a gap in my resume	Total length of any gaps in my resume in months.	...	I am on section 8 housing	How m times v hospital for y me illn
count	334.000000	334.000000	334.000000	334.000000	297.000000	334.000000	334.000000	334.000000	334.000000	334.000000	...	334.000000	334.000
mean	0.679641	0.239521	0.874251	0.077844	3.276094	0.098802	0.964072	0.110778	0.245509	8.497006	...	0.020958	1.19
std	0.467315	0.427431	0.332063	0.268328	14.126045	0.298844	0.186390	0.314328	0.431034	20.722643	...	0.143459	8.11
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000
25%	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	...	0.000000	0.000
50%	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	...	0.000000	0.000
75%	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	4.750000	...	0.000000	0.000
max	1.000000	1.000000	1.000000	1.000000	100.000000	1.000000	1.000000	1.000000	1.000000	100.000000	...	1.000000	100.000

8 rows × 25 columns

EDA I (Continued)

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 334 entries, 0 to 333
Data columns (total 31 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   I am currently employed at least part-time            334 non-null    int64
 1   I identify as having a mental illness                 334 non-null    int64
 2   Education                                             334 non-null    object
 3   I have my own computer separate from a smart phone    334 non-null    int64
 4   I have been hospitalized before for my mental illness 334 non-null    int64
 5   How many days were you hospitalized for your mental illness 297 non-null float64
 6   I am legally disabled                                 334 non-null    int64
 7   I have my regular access to the internet              334 non-null    int64
 8   I live with my parents                                334 non-null    int64
 9   I have a gap in my resume                             334 non-null    int64
 10  Total length of any gaps in my resume in months.      334 non-null    int64
 11  Annual income (including any social welfare programs) in USD 334 non-null int64
 12  I am unemployed                                       334 non-null    int64
 13  I read outside of work and school                     334 non-null    int64
 14  Annual income from social welfare programs            334 non-null    int64
 15  I receive food stamps                                 334 non-null    int64
 16  I am on section 8 housing                             334 non-null    int64
 17  How many times were you hospitalized for your mental illness 334 non-null int64
 18  Lack of concentration                                 333 non-null    float64
 19  Anxiety                                               334 non-null    int64
 20  Depression                                            334 non-null    int64
 21  Obsessive thinking                                    333 non-null    float64
 22  Mood swings                                           333 non-null    float64
 23  Panic attacks                                         333 non-null    float64
 24  Compulsive behavior                                   333 non-null    float64
 25  Tiredness                                             333 non-null    float64
 26  Age                                                   334 non-null    object
 27  Gender                                                334 non-null    object
 28  Household Income                                      334 non-null    object
 29  Region                                                332 non-null    object
 30  Device Type                                           334 non-null    object
dtypes: float64(7), int64(18), object(6)
memory usage: 81.0+ KB
```

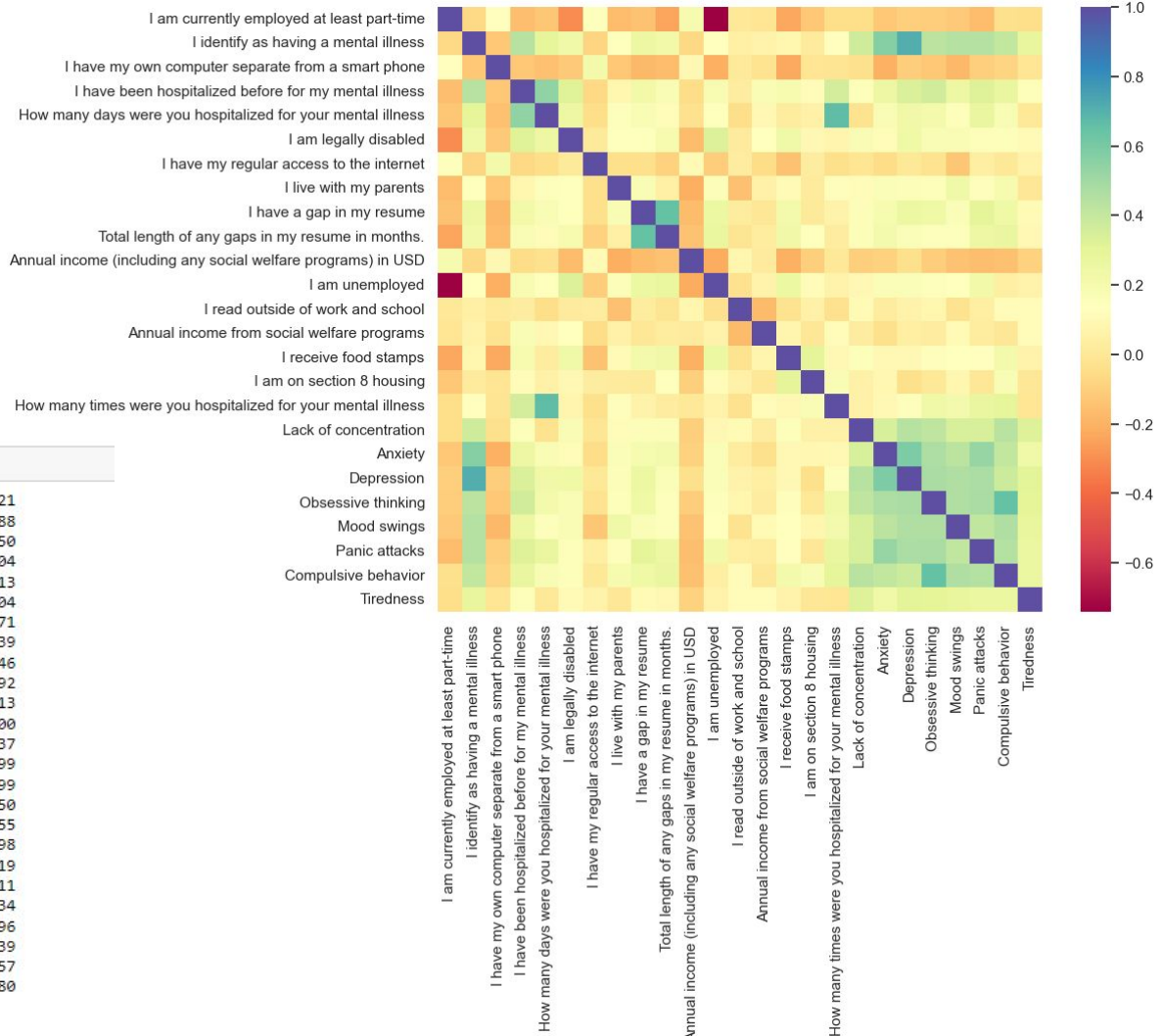
- There are 31 columns in total with 18 integers, 7 floats, and 6 objects.
- The difference of non-null count between counts show the need to deal with null values
- For the models to work properly, the object data types must either be converted to numericals or dropped completely

EDA II

- Correlation matrix helps us distinguish which columns has the strongest correlation with employment

```
corrmat['I am unemployed']
```

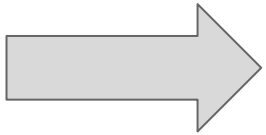
```
I am currently employed at least part-time                    -0.740321
I identify as having a mental illness                          0.134788
I have my own computer separate from a smart phone            -0.210350
I have been hospitalized before for my mental illness          0.186704
How many days were you hospitalized for your mental illness    0.141513
I am legally disabled                                          0.332804
I have my regular access to the internet                      -0.107071
I live with my parents                                         0.163039
I have a gap in my resume                                      0.252746
Total length of any gaps in my resume in months.               0.274092
Annual income (including any social welfare programs) in USD  -0.222513
I am unemployed                                                1.000000
I read outside of work and school                             -0.032137
Annual income from social welfare programs                     0.056499
I receive food stamps                                          0.257699
I am on section 8 housing                                      0.105050
How many times were you hospitalized for your mental illness   0.062755
Lack of concentration                                          0.091998
Anxiety                                                        0.168219
Depression                                                     0.170011
Obsessive thinking                                             0.147834
Mood swings                                                    0.133496
Panic attacks                                                  0.219739
Compulsive behavior                                            0.036757
Tiredness                                                      0.062480
Name: I am unemployed, dtype: float64
```



Preprocessing (Missing values)

```
df.isna().sum()
```

```
I am currently employed at least part-time                        0
I identify as having a mental illness                             0
Education                                                         0
I have my own computer separate from a smart phone                0
I have been hospitalized before for my mental illness             0
How many days were you hospitalized for your mental illness      37
I am legally disabled                                             0
I have my regular access to the internet                          0
I live with my parents                                            0
I have a gap in my resume                                         0
Total length of any gaps in my resume in months.                  0
Annual income (including any social welfare programs) in USD      0
I am unemployed                                                   0
I read outside of work and school                                 0
Annual income from social welfare programs                        0
I receive food stamps                                             0
I am on section 8 housing                                         0
How many times were you hospitalized for your mental illness      0
Lack of concentration                                             1
Anxiety                                                           0
Depression                                                        0
Obsessive thinking                                                1
Mood swings                                                       1
Panic attacks                                                     1
Compulsive behavior                                               1
Tiredness                                                         1
Age                                                               0
Gender                                                            0
Household Income                                                  0
Region                                                            2
Device Type                                                       0
dtype: int64
```



```
df1 = df.dropna()
```

```
df1.isna().sum()
```

```
I am currently employed at least part-time                        0
I identify as having a mental illness                             0
Education                                                         0
I have my own computer separate from a smart phone                0
I have been hospitalized before for my mental illness             0
How many days were you hospitalized for your mental illness      0
I am legally disabled                                             0
I have my regular access to the internet                          0
I live with my parents                                            0
I have a gap in my resume                                         0
Total length of any gaps in my resume in months.                  0
Annual income (including any social welfare programs) in USD      0
I am unemployed                                                   0
I read outside of work and school                                 0
Annual income from social welfare programs                        0
I receive food stamps                                             0
I am on section 8 housing                                         0
How many times were you hospitalized for your mental illness      0
Lack of concentration                                             0
Anxiety                                                           0
Depression                                                        0
Obsessive thinking                                                0
Mood swings                                                       0
Panic attacks                                                     0
Compulsive behavior                                               0
Tiredness                                                         0
Age                                                               0
Gender                                                            0
Household Income                                                  0
Region                                                            0
Device Type                                                       0
dtype: int64
```

- As presumed from the .info function earlier and confirmed with the code above, there are null values
- We then dropped all of the missing/null values

Preprocessing (Factorization of data)

- Columns comprised of strings must be factored into numerical categories
- Created new columns from the “levels” of the object/string columns

```
df['Education'].unique()
```

```
array(['High School or GED', 'Some Phd', 'Completed Undergraduate',
       'Some Undergraduate', 'Some\xa0Masters', 'Completed Masters',
       'Completed Phd', 'Some highschool'], dtype=object)
```

```
df.Gender.unique()
```

```
array(['Male', 'Female'], dtype=object)
```

```
df["Device Type"].unique()
```

```
array(['Android Phone / Tablet', 'MacOS Desktop / Laptop',
       'Windows Desktop / Laptop', 'iOS Phone / Tablet', 'Other'],
      dtype=object)
```

```
df.Region.unique()
```

```
array(['Mountain', 'East South Central', 'Pacific', 'New England',
       'East North Central', 'South Atlantic', 'Middle Atlantic',
       'West South Central', 'West North Central', nan], dtype=object)
```

```
df["Household Income"].unique()
```

```
array(['$25,000-$49,999', '$50,000-$74,999', '$150,000-$174,999',
       '$0-$9,999', '$100,000-$124,999', '$125,000-$149,999',
       'Prefer not to answer', '$10,000-$24,999', '$75,000-$99,999',
       '$200,000+', '$175,000-$199,999'], dtype=object)
```

```
df.Age.unique()
```

```
array(['30-44', '18-29', '45-60', '> 60'], dtype=object)
```

```
df1
```

Gender	Household Income	Region	Device Type	EducationNumerical	GenderNumerical	RegionNumerical	DeviceTypeNumerical	AgeNumerical	HouseIncEduNumerical
Male	25,000–49,999	Mountain	Android Phone / Tablet	0	1	0	0	0	1
Male	50,000–74,999	East South Central	MacOS Desktop / Laptop	1	1	1	1	1	2
Male	150,000–174,999	Pacific	MacOS Desktop / Laptop	2	1	2	1	0	4
Male	25,000–49,999	East North Central	iOS Phone / Tablet	2	1	3	2	0	1
Male	0–9,999	South Atlantic	Android Phone / Tablet	0	1	4	0	0	3
...	...	...	...	...	...	...	...	...	...
Female	25,000–49,999	Pacific	iOS Phone / Tablet	3	0	2	2	1	1
Female	50,000–74,999	Mountain	iOS Phone / Tablet	2	0	0	2	1	2
Male	50,000–74,999	Pacific	Windows Desktop / Laptop	3	1	2	3	1	2
Female	10,000–24,999	West North Central	Windows Desktop / Laptop	3	0	8	3	3	6
Female	0–9,999	West South Central	Android Phone / Tablet	3	0	7	0	1	3

Preprocessing (which columns are kept and gone)

- Input data for Logistic Regression model:
  - “I am legally disabled”
  - “Total length of any gaps in my resume in months.”
- Target data for Logistic Regression model:
  - “I am unemployed”
- Both of the input data had the highest correlation with the target data

```
#Dropping all columns except:
#I am legally disabled
#Total length of any gaps in my resume in months.

x = df1.drop(['I am unemployed', "Anxiety","Depression","Mood swings","I am currently employed at least part-time",
                "Education","I identify as having a mental illness","I have my own computer separate from a smart phone",
               "I have been hospitalized before for my mental illness","How many days were you hospitalized for your mental
            "How many times were you hospitalized for your mental illness",
                "I have my regular access to the internet","I live with my parents","I have a gap in my resume",
                "Tiredness","Age","Gender","Region","Device Type","Panic attacks","I am on section 8 housing",
               "I receive food stamps","Lack of concentration","Obsessive thinking","Household Income", "Compulsive behavic
                "I read outside of work and school",
                "Annual income from social welfare programs","Annual income (including any social welfare programs) in USD"
            "HouseIncEduNumerical","AgeNumerical","DeviceTypeNumerical","RegionNumerical","GenderNumerical","EducationNumer
               ],axis=1)

y = df1["I am unemployed"].values.reshape(-1,1) #Target is whether they are unemployed or not
```

Preprocessing (which columns are kept and gone) Continued

- Input variables for Classification Algorithms (Decision Tree, KNN, Random Forest, Baseline Random Forest):
  - All of the columns except the six columns with strings/described as objects
- Target variable for Classification Algorithms (Decision Tree, KNN, Random Forest, Baseline Random Forest):
  - “I am unemployed”

```
#Keep all columns except those that include strings
#Education, gender, region, device type, age, and household income are still incorperated into model because we have already
#factored their contents into newly created columns
x1 = df1.drop(['I am unemployed',"Education","Gender","Region","Device Type","Age",
               "Annual income (including any social welfare programs) in USD",
               "Household Income"
                 ],axis=1)
y1 = df1["I am unemployed"].values.reshape(-1,1) #Target is whether they are unemployed or not
```

Models (Logistic Regression)

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

```
from sklearn.linear_model import LogisticRegression

model_clf = LogisticRegression()
```

```
model_clf.fit(x_train, y_train)
```

```
C:\Users\jamir\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
sed when a 1d array was expected. Please change the shape of y to (n_samp
  y = column_or_1d(y, warn=True)
```

```
LogisticRegression()
```

First step: Splitting the data into the training and testing sets
Second step: Construct the model
Third step: Fit the training data set into the model

```
x_train
```

	I am legally disabled	Total length of any gaps in my resume in months.
135	0	0
245	0	0
112	0	0
22	0	0
191	0	3
...	...	...
214	0	0
82	0	0
122	0	0
307	0	0
118	0	0

205 rows × 2 columns

Models (Decision Tree)

- Will be the training and test dataset for Decision Tree, KNN, Random Forest, a Baseline Random Forest models

```
x_train1, x_test1, y_train1, y_test1 = train_test_split(x1, y1, test_size=0.3, random_state=42)
```

```
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
```

```
clf = DecisionTreeClassifier()
#criterion="entropy", max_depth=3

# Train Decision Tree Classifer
clf = clf.fit(x_train1, y_train1)

#Predict the response for test dataset
y_pred = clf.predict(x_test1)
```

More Optomized Version

```
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)

# Train Decision Tree Classifer
clf = clf.fit(x_train1, y_train1)

#Predict the response for test dataset
y_pred = clf.predict(x_test1)
```

- Using DecisionTreeClassifier instead of DecisionTreeRegressor because we have a classification problem

- Max depth = 3 limits the tree to only three branches
- Criterion = entropy calculates how distinct the elements are from the target

Models (K-nearest Neighbor)

KNN

```
from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=3)
```

```
#Train
knn_model.fit(x_train1, y_train1)
```

Models (Random Forest)

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.ensemble import AdaBoostClassifier
```

```
regr_1 = DecisionTreeClassifier(max_depth=5)
```

```
regr_2 = AdaBoostClassifier(DecisionTreeClassifier(max_depth=5), n_estimators=50 )
```

```
regr_1.fit(x_train1,y_train1)
```

```
DecisionTreeClassifier(max_depth=5)
```

```
regr_2.fit(x_train1,y_train1)
```

```
C:\Users\jamir\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataCon
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), fo
  y = column_or_1d(y, warn=True)
```

```
AdaBoostClassifier(base_estimator=DecisionTreeClassifier(max_depth=5))
```

Models (Baseline Random Forest and Updated Random Forest)

```
from sklearn.ensemble import RandomForestClassifier
```

```
randfor = RandomForestClassifier(max_depth=5)
```

```
randfor.fit(x_train1,y_train1)
```

```
C:\Users\jamir\AppData\Local\Temp\ipykernel_22536\3549735244.py:1: D
a 1d array was expected. Please change the shape of y to (n_samples,
  randfor.fit(x_train1,y_train1)
```

```
RandomForestClassifier(max_depth=5)
```

```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(n_estimators=50, max_features="auto", random_state=44)
rf_model.fit(x_train1, y_train1)
```

```
predictions = rf_model.predict(x_test1)
predictions
```

```
array([1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0], dtype=int64)
```

Model (Tweaking the updated Random Forest model)

Minimizing the nodes/n_estimators, the worse the model performs. However, increasing the nodes/n_estimators, the better the model performs.

```
from sklearn.ensemble import RandomForestClassifier
rf_model1 = RandomForestClassifier(n_estimators=1, max_features="auto", random_state=44)
rf_model1.fit(x_train1, y_train1)
```

```
C:\Users\jamir\AppData\Local\Temp\ipykernel_22536\509835577.py:3: DataConversionWarning: A
1d array was expected. Please change the shape of y to (n_samples,), for example using rave
  rf_model1.fit(x_train1, y_train1)
```

```
RandomForestClassifier(n_estimators=1, random_state=44)
```

```
predictions1 = rf_model1.predict(x_test1)
```

Reports on predictions (Logistic regression)

```
model_clf.score(x_train, y_train)
```

```
0.7707317073170732
```

```
model_clf.score(x_test, y_test)
```

```
0.7640449438202247
```

```
model_clf.coef_
```

```
array([[1.24820875, 0.01938461]])
```

```
model_clf.intercept_
```

```
array([-1.40201585])
```

	precision	recall	f1-score	support
Employed	0.78	0.97	0.86	151
Unemployed	0.71	0.22	0.34	54
accuracy			0.77	205
macro avg	0.74	0.59	0.60	205
weighted avg	0.76	0.77	0.72	205

Reports on predictions (Decision Tree)

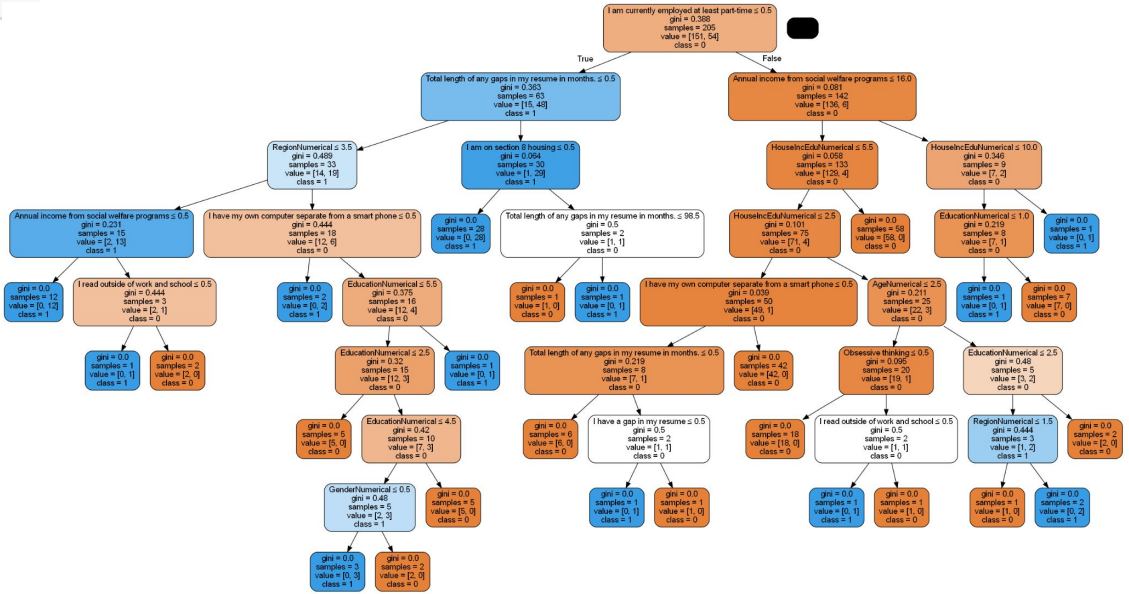
```
clf.score(x_train1, y_train1)
```

```
1.0
```

```
clf.score(x_test1, y_test1)
```

```
0.8089887640449438
```

- Due to the perfect training score, this is a case of overfitting
- In order to improve our Decision Tree, we must optimize our model



Reports on predictions (Decision Tree) Continued

Optimized version

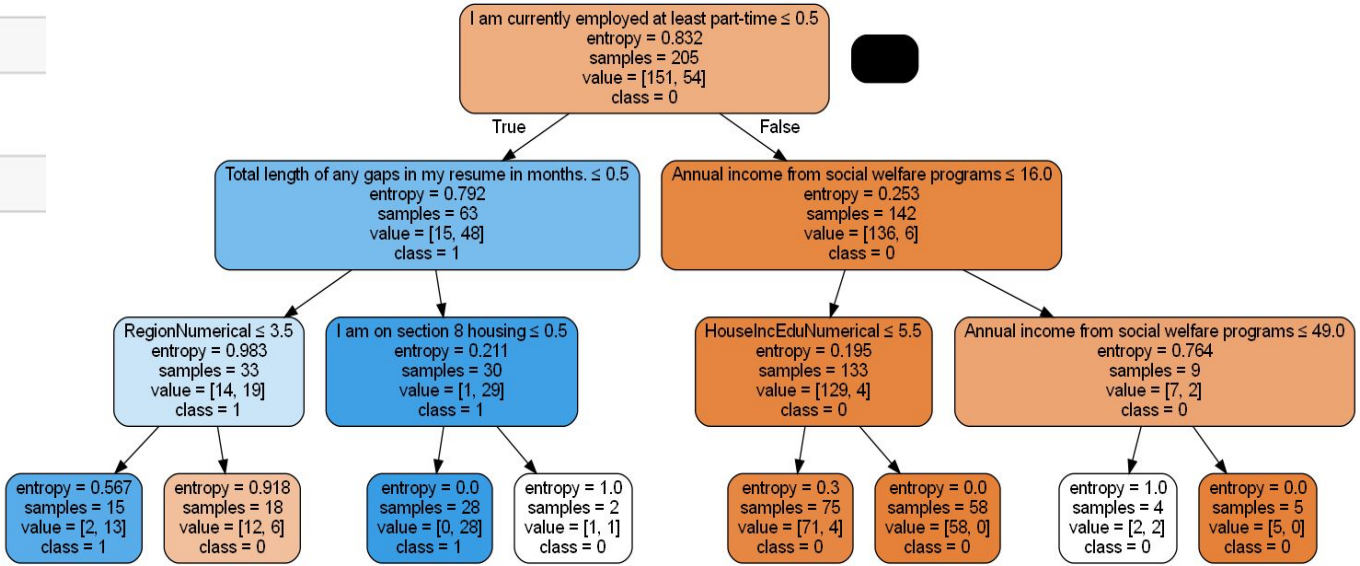
```
clf.score(x_train1, y_train1)
```

```
0.926829268292683
```

```
clf.score(x_test1, y_test1)
```

```
0.8651685393258427
```

- More of a balanced Decision Tree model as both training and test scores are highly accurate without being unreasonable



Reports on predictions (KNN Model)

	precision	recall	f1-score	support
0	0.83	0.95	0.89	151
1	0.78	0.46	0.58	54
accuracy			0.82	205
macro avg	0.81	0.71	0.74	205
weighted avg	0.82	0.82	0.81	205

	precision	recall	f1-score	support
0	0.77	0.89	0.83	66
1	0.42	0.22	0.29	23
accuracy			0.72	89
macro avg	0.59	0.56	0.56	89
weighted avg	0.68	0.72	0.69	89

The first classification report is for the training data whereas the second report is for the testing data split.

Reports on predictions (Boosted Decision Tree and Baseline Random Forest)

```
regr_1.score(x_test1,y_test1)
```

```
0.8314606741573034
```

```
regr_2.score(x_test1,y_test1)
```

```
0.8314606741573034
```

Baseline random forest

```
randfor.score(x_test1,y_test1)
```

```
0.8202247191011236
```

Reports on predictions (Updated Random Forest Tree)

```
rf_model.score(x_train1,y_train1)
```

```
1.0
```

```
rf_model.score(x_test1,y_test1)
```

```
0.8876404494382022
```

```
print(classification_report(y_test1,predictions))
```

	precision	recall	f1-score	support
0	0.92	0.92	0.92	66
1	0.78	0.78	0.78	23
accuracy			0.89	89
macro avg	0.85	0.85	0.85	89
weighted avg	0.89	0.89	0.89	89

Reports on predictions (Tweaking the updated Random Forest model)

```
rf_model1.score(x_train1,y_train1)
```

```
0.9463414634146341
```

```
rf_model1.score(x_test1,y_test1)
```

```
0.8202247191011236
```

```
print(classification_report(y_test1,predictions1))
```

```
              precision    recall  f1-score   support

           0       0.93      0.82      0.87        66
           1       0.61      0.83      0.70        23

    accuracy                           0.82        89
   macro avg       0.77      0.82      0.79        89
weighted avg       0.85      0.82      0.83        89
```