Homeless Youth Dataset Model

Aman Gangwani, Owen Finkbeiner

AUC score - .99 with 10 fold cross validation.

In [1]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

Introduction

The goal of this project was to write a multi-class classfiication model on the homeless youth dataset.The task was to predict what kind of drug the homeless youth are at risk of using and whether they do not use any drugs. We don't have any test set so our goal was to maximize the 10 fold cross validation. The dataset provided had a ton of different features. We selected these features mostly by just going through the data description file to see what we thought would be most useful in predicting these values.

In [2]:

```
data = pd.read_csv("project_4_data.csv")
clean_data = pd.DataFrame()
```

```
/Users/amangangwani/miniconda3/lib/python3.8/site-packages/IPython/core/interactiveshell.
py:3145: DtypeWarning: Columns (2,6,9,21,26,76,279,371,458,474,493,497,499,945,946) have
mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [3]:

```
data = data[data['selfhomeless'] == 2]
# we only want the people that are homeless
```

In [4]:

```
data.head()
```

Out[4]:

	pid	screen1_sleep	screen1_sleep_18_text	screen_voucher	screen2_long	screen3_age	realm_score	consent	consent_a
10	1006	15	.	NaN	NaN	22.0	9	1.0	
11	1006	15	.	NaN	NaN	22.0	9	1.0	
12	1006	15	.	NaN	NaN	22.0	9	1.0	
13	1006	15	.	NaN	NaN	22.0	9	1.0	
14	1006	15	.	NaN	NaN	22.0	9	1.0	

5 rows × 1210 columns

In [5]:

```
clean_data['pid'] = data['pid']
```

In [6]:

```
data['screen1_sleep'].value_counts() # here we're basically turning this into one hot enc
```

```
oding
# Shelter: 8 or 9
# Insecure: 15, 16, or 17
# Secure: all others
```

Out[6]:

```
8     1520
15    1320
5      405
9      360
17     275
13     230
16     205
10     120
1      115
18      70
3       45
6       40
11      25
7       15
14      15
12      15
2       10
Name: screen1_sleep, dtype: int64
```

In [7]:

```
shelter = []
insecure = []
secure = []

shelter_ = 0
insecure_ = 0
secure_ = 0

for i in data['screen1_sleep']:
    if i == 8 or i == 9:
        shelter_ = 1
    elif i == 15 or i == 16 or i == 17:
        insecure_ = 1
    else:
        secure_ = 1
    shelter.append(shelter_)
    insecure.append(insecure_)
    secure.append(secure_)

clean_data['screen1_sleep_shelter'] = shelter
clean_data['screen1_sleep_insecure'] = insecure
clean_data['screen1_sleep_secure'] = secure
#we're basically creating dummy variables here for whatever we may need
```

In [8]:

```
clean_data['screen3_age'] = data['screen3_age']
```

In [9]:

```
clean_data['screen3_age'] = clean_data['screen3_age'].fillna(21)
```

In [10]:

```
data['realm_score'].value_counts()
```

Out[10]:

```
9     1530
.      815
4      505
9      440
8      320
8      240
```

```
7       213
7       207
6       125
6       105
5        75
4        65
5        65
0        15
10       15
0        15
2        10
3         5
1         5
3         5
1         5
Name: realm_score, dtype: int64
```

In [11]:

```
help_needed = []

for i in data['realm_score']:
    if isinstance(i, int):
        if i < 4:
            help_ = 1
        else:
            help_ = 0
    else:
        help_ = 0
    help_needed.append(help_)

clean_data['realm_score_help_needed'] = help_needed
#turning this into one hot encoding
```

In [12]:

```
# we're just going to be filling na values with 0 since there aren't many missing for eac
h of these
data['genderidentity_3'] = data['genderidentity_3'].fillna(0)
data['genderidentity_4'] = data['genderidentity_4'].fillna(0)
data['genderidentity_5'] = data['genderidentity_5'].fillna(0)
data['genderidentity_6'] = data['genderidentity_6'].fillna(0)
clean_data['gender_identity_lgbt'] = data['genderidentity_3'] + data['genderidentity_4']
+ data['genderidentity_5'] + data['genderidentity_6']
```

In [13]:

```
clean_data['birthsex_male'] = data['birthsex']
clean_data['birthsex_male'] = clean_data['birthsex_male'].replace(2, 0)
clean_data['birthsex_male'] = clean_data['birthsex_male'].fillna(0)
# we're replacinga. few values and then filling na values ofr the rest
```

In [14]:

```
lgbt = []

for i in data['sexualorientation']:
    if i == 2.0 or i == 5.0:
        lgbt_ = 0
    else:
        lgbt_ = 1
    lgbt.append(lgbt_)

clean_data['sexualorientation_lgbt'] = lgbt
#again if the value is 2 or 5 where 2 is straight and 5 is i don't know, we're turning th
ose into 0 and then
# otherwise we're keeping the values as a 1 basically it's Yes are you lgbt or not.
```

In [15]:

```
clean_data['first_homeless'] = data['firsthomeless_1'].fillna(clean_data['screen3_age'])
```

```
clean_data['years_since_first_homeless'] = clean_data['screen3_age'] - clean_data['first_
homeless']
```

In [16]:

```
data['reasonhomeless_1'].value_counts()
```

Out[16]:

```
1.0    2520
0.0    1060
Name: reasonhomeless_1, dtype: int64
```

In [17]:

```
clean_data['education'] = data['education'].replace(-99,1)
clean_data['education'] = clean_data['education'].fillna(1)
```

In [18]:

```
data['inschool'].value_counts()
```

Out[18]:

```
1.0    3985
2.0     800
Name: inschool, dtype: int64
```

In [19]:

```
clean_data['inschool'] = data['inschool'] - 1
```

In [20]:

```
data['ttlfcplacements_1'].value_counts()
```

Out[20]:

```
 1.0     350
 2.0     235
 3.0     205
 4.0     165
 25.0    145
 5.0     105
 7.0      95
 10.0     85
 8.0      65
 13.0     45
 6.0      40
 14.0     40
 11.0     35
 15.0     35
 12.0     35
 18.0     30
 16.0     25
 9.0      25
 17.0     15
 23.0     15
 20.0      5
 22.0      5
 24.0      5
-99.0      5
 21.0      5
Name: ttlfcplacements_1, dtype: int64
```

In [21]:

```
data['ttlfcplacements_1'].isna().sum()
```

Out[21]:

```
2970
```

In [22]:

```
clean_data['foster_care_placements'] = data['ttlfcplacements_1'].replace(-99,0)
clean_data['foster_care_placements'] = clean_data['foster_care_placements'].replace(np.n
an,0)
```

In [23]:

```
data['reasonhomeless_5'] = data['reasonhomeless_5'].fillna(0)
data['reasonhomeless_6'] = data['reasonhomeless_6'].fillna(0)
data['reasonhomeless_7'] = data['reasonhomeless_7'].fillna(0)
data['reasonhomeless_8'] = data['reasonhomeless_8'].fillna(0)

clean_data['reason_homeless_ran_away'] = data['reasonhomeless_5'] + data['reasonhomeless_
6'] + data['reasonhomeless_7'] + data['reasonhomeless_8']
```

In [24]:

```
clean_data['reason_homeless_ran_away'] = clean_data['reason_homeless_ran_away'].replace(
2, 1)
clean_data['reason_homeless_ran_away'] = clean_data['reason_homeless_ran_away'].replace(
3, 1)
clean_data['reason_homeless_ran_away'] = clean_data['reason_homeless_ran_away'].replace(
4, 1)
clean_data['reason_homeless_ran_away'].value_counts()
```

Out[24]:

```
0.0    3975
1.0     810
Name: reason_homeless_ran_away, dtype: int64
```

In [25]:

```
clean_data['jjinvolve'] = data['jjinvolve'] - 1
clean_data['jjinvolve'] = clean_data['jjinvolve'].fillna(0)
```

In [26]:

```
clean_data['everarrest'] = data['everarrest'] - 1
```

In [27]:

```
clean_data['jail_homeless'] = data['jail_homeless'] - 1
clean_data['jail_homeless'] = clean_data['jail_homeless'].fillna(0)
```

In [28]:

```
clean_data['incomegen_12mo_1'] = data['incomegen_12mo_1'] - 1
clean_data['incomegen_12mo_2'] = data['incomegen_12mo_2'] - 1
clean_data['incomegen_12mo_3'] = data['incomegen_12mo_3'] - 1
clean_data['incomegen_12mo_4'] = data['incomegen_12mo_4'] - 1
clean_data['incomegen_12mo_5'] = data['incomegen_12mo_5'] - 1
clean_data['incomegen_12mo_6'] = data['incomegen_12mo_6'] - 1
clean_data['incomegen_12mo_7'] = data['incomegen_12mo_7'] - 1
clean_data['incomegen_12mo_8'] = data['incomegen_12mo_8'] - 1
clean_data['incomegen_12mo_9'] = data['incomegen_12mo_9'] - 1
clean_data['incomegen_12mo_10'] = data['incomegen_12mo_10'] - 1
clean_data['incomegen_12mo_11'] = data['incomegen_12mo_11'] - 1
clean_data['incomegen_12mo_12'] = data['incomegen_12mo_12'] - 1
clean_data['incomegen_12mo_13'] = data['incomegen_12mo_13'] - 1
clean_data['incomegen_12mo_14'] = data['incomegen_12mo_14'] - 1
clean_data['incomegen_12mo_15'] = data['incomegen_12mo_15'] - 1
clean_data['incomegen_12mo_16'] = data['incomegen_12mo_16'] - 1
clean_data['incomegen_12mo_17'] = data['incomegen_12mo_17'] - 1
```

In [29]:

```
clean_data['incomegen_12mo_1'] = clean_data['incomegen_12mo_1'].fillna(0)
clean_data['incomegen_12mo_2'] = clean_data['incomegen_12mo_2'].fillna(0)
```

```
clean_data['incomegen_12mo_3'] = clean_data['incomegen_12mo_3'].fillna(0)
clean_data['incomegen_12mo_4'] = clean_data['incomegen_12mo_4'].fillna(0)
clean_data['incomegen_12mo_5'] = clean_data['incomegen_12mo_5'].fillna(0)
clean_data['incomegen_12mo_6'] = clean_data['incomegen_12mo_6'].fillna(0)
clean_data['incomegen_12mo_7'] = clean_data['incomegen_12mo_7'].fillna(0)
clean_data['incomegen_12mo_8'] = clean_data['incomegen_12mo_8'].fillna(0)
clean_data['incomegen_12mo_9'] = clean_data['incomegen_12mo_9'].fillna(0)
clean_data['incomegen_12mo_10'] = clean_data['incomegen_12mo_10'].fillna(0)
clean_data['incomegen_12mo_11'] = clean_data['incomegen_12mo_11'].fillna(0)
clean_data['incomegen_12mo_12'] = clean_data['incomegen_12mo_12'].fillna(0)
clean_data['incomegen_12mo_13'] = clean_data['incomegen_12mo_13'].fillna(0)
clean_data['incomegen_12mo_14'] = clean_data['incomegen_12mo_14'].fillna(0)
clean_data['incomegen_12mo_15'] = clean_data['incomegen_12mo_15'].fillna(0)
clean_data['incomegen_12mo_16'] = clean_data['incomegen_12mo_16'].fillna(0)
clean_data['incomegen_12mo_17'] = clean_data['incomegen_12mo_17'].fillna(0)
```

In [30]:

```
clean_data['working'] = data['working'] - 1
clean_data['working'] = clean_data['working'].fillna(0)
```

In [31]:

```
clean_data['ttllegalhours'] = data['ttllegalhours_1'].fillna(0)
```

In [32]:

```
clean_data['sex_ever'] = data['sex_ever'] - 1
clean_data['sex_ever'] = clean_data['sex_ever'].fillna(0)
```

In [33]:

```
clean_data['sex_3mo'] = data['sex_3mo'] - 1
clean_data['sex_3mo'] = clean_data['sex_3mo'].fillna(0)
```

In [34]:

```
clean_data['num_sexpart_3mo'] = data['num_sexpart_3mo']
clean_data['num_sexpart_3mo'] = clean_data['num_sexpart_3mo'].fillna(0)
```

In [35]:

```
clean_data['condom_use_3mo_1'] = data['condom_use_3mo_1'].fillna(1)
```

In [36]:

```
clean_data['last_sui'] = data['last_sui'] - 1
clean_data['last_sui'] = clean_data['last_sui'].fillna(0)
```

In [37]:

```
clean_data['ever_sextrade'] = data['ever_sextrade'] - 1
clean_data['ever_sextrade'] = clean_data['ever_sextrade'].fillna(0)
```

In [38]:

```
clean_data['ever_sextradetraffic'] = data['ever_sextradetraffic'] - 1
clean_data['ever_sextradetraffic'] = clean_data['ever_sextradetraffic'].fillna(0)
```

In [39]:

```
clean_data['hpv_2'] = data['hpv_2'].fillna(0)
clean_data['hpv_2'] = clean_data['hpv_2'].replace(-99,0)
```

In [40]:

```
std = []

for i in data['std_status']:
    if i == 2:
        std_ = 1
    else:
        std_ = 0
    std.append(std_)

clean_data['std_pos'] = std

# do you have an std or not is what we're turning this feature into
```

In [41]:

```
hepC = []

for i in data['hepc_status']:
    if i == 2:
        hepc_ = 1
    else:
        hepc_ = 0
    hepC.append(hepc_)

clean_data['hepC_pos'] = hepC

# do you have hepc or not is what this is being transformed into
```

In [42]:

```
clean_data['preg_numtimes'] = data['preg_numtimes'] - 1
clean_data['preg_numtimes'] = clean_data['preg_numtimes'].fillna(0)
```

In [43]:

```
clean_data['preg_numunplan'] = data['preg_numunplan'] - 1
clean_data['preg_numunplan'] = clean_data['preg_numunplan'].fillna(0)
```

In [44]:

```
clean_data['preg_numchildliving'] = data['preg_numchildliving'] - 1
clean_data['preg_numchildliving'] = clean_data['preg_numchildliving'].fillna(0)
```

In [45]:

```
clean_data['ace_emotabuse'] = data['ace_emotabuse'] - 1
clean_data['ace_physicalabuse'] = data['ace_physicalabuse'] - 1
clean_data['ace_sexualabuse'] = data['ace_sexualabuse'] - 1
clean_data['ace_emotneglect'] = data['ace_emotneglect'] - 1
clean_data['ace_physneglect'] = data['ace_physneglect'] - 1
clean_data['ace_divorce'] = data['ace_divorce'] - 1
clean_data['ace_domesticviol'] = data['ace_domesticviol'] - 1
clean_data['ace_caregiversubstan'] = data['ace_caregiversubstan'] - 1
clean_data['ace_caregiverdepress'] = data['ace_caregiverdepress'] - 1
clean_data['ace_caregiverincar'] = data['ace_caregiverincar'] - 1
```

In [46]:

```
clean_data['ace_emotabuse'] = clean_data['ace_emotabuse'].fillna(0)
clean_data['ace_physicalabuse'] = clean_data['ace_physicalabuse'].fillna(0)
clean_data['ace_sexualabuse'] = clean_data['ace_sexualabuse'].fillna(1)
clean_data['ace_emotneglect'] = clean_data['ace_emotneglect'].fillna(0)
clean_data['ace_physneglect'] = clean_data['ace_physneglect'].fillna(0)
clean_data['ace_divorce'] = clean_data['ace_divorce'].fillna(0)
clean_data['ace_domesticviol'] = clean_data['ace_domesticviol'].fillna(0)
clean_data['ace_caregiversubstan'] = clean_data['ace_caregiversubstan'].fillna(0)
clean_data['ace_caregiverdepress'] = clean_data['ace_caregiverdepress'].fillna(0)
clean_data['ace_caregiverincar'] = clean_data['ace_caregiverincar'].fillna(0)
```

In [47]:

```
clean_data['vict_robbery'] = data['vict_robbery'] - 1
clean_data['vict_assltwweapon'] = data['vict_assltwweapon'] - 1
```

```
clean_data['vict_assaultwoweapon'] = data['vict_assaultwoweapon'] - 1
clean_data['vict_threatenassault'] = data['vict_threatenassault'] - 1

clean_data['vict_biasattack_1'] = data['vict_biasattack_1'] - 1
clean_data['vict_biasattack_2'] = data['vict_biasattack_2'] - 1
clean_data['vict_biasattack_3'] = data['vict_biasattack_3'] - 1
clean_data['vict_biasattack_4'] = data['vict_biasattack_4'] - 1
clean_data['vict_biasattack_5'] = data['vict_biasattack_5'] - 1
clean_data['vict_biasattack_6'] = data['vict_biasattack_6'] - 1
clean_data['vict_biasattack_7'] = data['vict_biasattack_7'] - 1

clean_data['vict_gang'] = data['vict_gang'] - 1
clean_data['vict_witness'] = data['vict_witness'] - 1
```

In [48]:

```
clean_data['vict_robbery'] = clean_data['vict_robbery'].fillna(0)
clean_data['vict_assltwweapon'] = clean_data['vict_assltwweapon'].fillna(0)
clean_data['vict_assaultwoweapon'] = clean_data['vict_assaultwoweapon'].fillna(0)
clean_data['vict_threatenassault'] = clean_data['vict_threatenassault'].fillna(0)

clean_data['vict_biasattack_1'] = clean_data['vict_biasattack_1'].fillna(0)
clean_data['vict_biasattack_2'] = clean_data['vict_biasattack_2'].fillna(0)
clean_data['vict_biasattack_3'] = clean_data['vict_biasattack_3'].fillna(0)
clean_data['vict_biasattack_4'] = clean_data['vict_biasattack_4'].fillna(0)
clean_data['vict_biasattack_5'] = clean_data['vict_biasattack_5'].fillna(0)
clean_data['vict_biasattack_6'] = clean_data['vict_biasattack_6'].fillna(0)
clean_data['vict_biasattack_7'] = clean_data['vict_biasattack_7'].fillna(0)

clean_data['vict_gang'] = clean_data['vict_gang'].fillna(0)
clean_data['vict_witness'] = clean_data['vict_witness'].fillna(0)
```

In [49]:

```
ipv_vic = []

for i in data['vict_ipv_vic']:
    if i == 5:
        ipv_vic_ = 1
    else:
        ipv_vic_ = 0
    ipv_vic.append(ipv_vic_)

clean_data['vict_ipv_vic'] = ipv_vic

# were you a victim of an abusive relationship
```

In [50]:

```
ipv_perp = []

for i in data['vict_ipv_perp']:
    if i == 5:
        ipv_perp_ = 1
    else:
        ipv_perp_ = 0
    ipv_perp.append(ipv_perp_)

clean_data['vict_ipv_perp'] = ipv_perp
# were you the perpetrator in an abusive relationship?
```

In [51]:

```
clean_data['vict_sexlasslt'] = data['vict_sexlasslt']
clean_data['vict_forcesex'] = data['vict_forcesex']
clean_data['vict_sexlassltexam'] = data['vict_sexlassltexam']
```

In [52]:

```
clean_data['vict_sexlasslt'] = clean_data['vict_sexlasslt'].fillna(0)
clean_data['vict_forcesex'] = clean_data['vict_forcesex'].fillna(0)
```

```
clean_data['vict_sexlassltexam'] = clean_data['vict_sexlassltexam'].fillna(0)
```

In [53]:

```
clean_data['cope_1'] = data['cope_1']
clean_data['cope_2'] = data['cope_2']
clean_data['cope_3'] = data['cope_3']
clean_data['cope_4'] = data['cope_4']
clean_data['cope_5'] = data['cope_5']
clean_data['cope_6'] = data['cope_6']
clean_data['cope_7'] = data['cope_7']
clean_data['cope_8'] = data['cope_8']
clean_data['cope_9'] = data['cope_9']
clean_data['cope_10'] = data['cope_10']
clean_data['cope_11'] = data['cope_11']
clean_data['cope_12'] = data['cope_12']
clean_data['cope_13'] = data['cope_13']
clean_data['cope_14'] = data['cope_14']
```

In [54]:

```
clean_data['cope_1'] = clean_data['cope_1'].fillna(1)
clean_data['cope_2'] = clean_data['cope_2'].fillna(1)
clean_data['cope_3'] = clean_data['cope_3'].fillna(1)
clean_data['cope_4'] = clean_data['cope_4'].fillna(1)
clean_data['cope_5'] = clean_data['cope_5'].fillna(1)
clean_data['cope_6'] = clean_data['cope_6'].fillna(1)
clean_data['cope_7'] = clean_data['cope_7'].fillna(1)
clean_data['cope_8'] = clean_data['cope_8'].fillna(1)
clean_data['cope_9'] = clean_data['cope_9'].fillna(1)
clean_data['cope_10'] = clean_data['cope_10'].fillna(1)
clean_data['cope_11'] = clean_data['cope_11'].fillna(1)
clean_data['cope_12'] = clean_data['cope_12'].fillna(1)
clean_data['cope_13'] = clean_data['cope_13'].fillna(1)
clean_data['cope_14'] = clean_data['cope_14'].fillna(1)
```

In [55]:

```
clean_data['descrim_1'] = data['descrim_1']
clean_data['descrim_2'] = data['descrim_2']
clean_data['descrim_3'] = data['descrim_3']
clean_data['descrim_4'] = data['descrim_4']
clean_data['descrim_5'] = data['descrim_5']
```

In [56]:

```
clean_data['descrim_1'] = clean_data['descrim_1'].fillna(1)
clean_data['descrim_2'] = clean_data['descrim_2'].fillna(1)
clean_data['descrim_3'] = clean_data['descrim_3'].fillna(1)
clean_data['descrim_4'] = clean_data['descrim_4'].fillna(1)
clean_data['descrim_5'] = clean_data['descrim_5'].fillna(1)
```

In [57]:

```
clean_data['discrim_reasons_1'] = data['discrim_reasons_1']
clean_data['discrim_reasons_2'] = data['discrim_reasons_2']
clean_data['discrim_reasons_3'] = data['discrim_reasons_3']
clean_data['discrim_reasons_4'] = data['discrim_reasons_4']
clean_data['discrim_reasons_5'] = data['discrim_reasons_5']
clean_data['discrim_reasons_6'] = data['discrim_reasons_6']
clean_data['discrim_reasons_7'] = data['discrim_reasons_7']
clean_data['discrim_reasons_8'] = data['discrim_reasons_8']
clean_data['discrim_reasons_9'] = data['discrim_reasons_9']
clean_data['discrim_reasons_10'] = data['discrim_reasons_10']
clean_data['discrim_reasons_11'] = data['discrim_reasons_11']
clean_data['discrim_reasons_12'] = data['discrim_reasons_12']
```

In [58]:

```
clean_data['discrim_reasons_1'] = clean_data['discrim_reasons_1'].fillna(0)
```

```
clean_data['discrim_reasons_2'] = clean_data['discrim_reasons_2'].fillna(0)
clean_data['discrim_reasons_3'] = clean_data['discrim_reasons_3'].fillna(0)
clean_data['discrim_reasons_4'] = clean_data['discrim_reasons_4'].fillna(0)
clean_data['discrim_reasons_5'] = clean_data['discrim_reasons_5'].fillna(0)
clean_data['discrim_reasons_6'] = clean_data['discrim_reasons_6'].fillna(0)
clean_data['discrim_reasons_7'] = clean_data['discrim_reasons_7'].fillna(0)
clean_data['discrim_reasons_8'] = clean_data['discrim_reasons_8'].fillna(0)
clean_data['discrim_reasons_9'] = clean_data['discrim_reasons_9'].fillna(0)
clean_data['discrim_reasons_10'] = clean_data['discrim_reasons_10'].fillna(0)
clean_data['discrim_reasons_11'] = clean_data['discrim_reasons_11'].fillna(0)
clean_data['discrim_reasons_12'] = clean_data['discrim_reasons_12'].fillna(0)
```

In [59]:

```
clean_data['stress_streets_4'] = data['stress_streets_4']
clean_data['stress_streets_6'] = data['stress_streets_6']
clean_data['stress_streets_7'] = data['stress_streets_7']
clean_data['stress_streets_8'] = data['stress_streets_8']
clean_data['stress_streets_9'] = data['stress_streets_9']
clean_data['stress_streets_10'] = data['stress_streets_10']
clean_data['stress_streets_11'] = data['stress_streets_11']
clean_data['stress_streets_12'] = data['stress_streets_12']
clean_data['stress_streets_14'] = data['stress_streets_14']
clean_data['stress_streets_15'] = data['stress_streets_15']
clean_data['stress_streets_16'] = data['stress_streets_16']
clean_data['stress_streets_17'] = data['stress_streets_17']
clean_data['stress_streets_18'] = data['stress_streets_18']
clean_data['stress_streets_19'] = data['stress_streets_19']
clean_data['stress_streets_20'] = data['stress_streets_20']
```

In [60]:

```
clean_data['stress_streets_4'] = clean_data['stress_streets_4'].fillna(1)
clean_data['stress_streets_6'] = clean_data['stress_streets_6'].fillna(1)
clean_data['stress_streets_7'] = clean_data['stress_streets_7'].fillna(1)
clean_data['stress_streets_8'] = clean_data['stress_streets_8'].fillna(1)
clean_data['stress_streets_9'] = clean_data['stress_streets_9'].fillna(1)
clean_data['stress_streets_10'] = clean_data['stress_streets_10'].fillna(1)
clean_data['stress_streets_11'] = clean_data['stress_streets_11'].fillna(1)
clean_data['stress_streets_12'] = clean_data['stress_streets_12'].fillna(1)
clean_data['stress_streets_14'] = clean_data['stress_streets_14'].fillna(1)
clean_data['stress_streets_15'] = clean_data['stress_streets_15'].fillna(1)
clean_data['stress_streets_16'] = clean_data['stress_streets_16'].fillna(1)
clean_data['stress_streets_17'] = clean_data['stress_streets_17'].fillna(1)
clean_data['stress_streets_18'] = clean_data['stress_streets_18'].fillna(1)
clean_data['stress_streets_19'] = clean_data['stress_streets_19'].fillna(1)
clean_data['stress_streets_20'] = clean_data['stress_streets_20'].fillna(1)
```

In [61]:

```
clean_data['mindfulness_1'] = data['mindfulness_1']
clean_data['mindfulness_4'] = data['mindfulness_4']
clean_data['mindfulness_5'] = data['mindfulness_5']
clean_data['mindfulness_6'] = data['mindfulness_6']
clean_data['mindfulness_7'] = data['mindfulness_7']
clean_data['mindfulness_8'] = data['mindfulness_8']
```

In [62]:

```
clean_data['mindfulness_1'] = clean_data['mindfulness_1'].fillna(1)
clean_data['mindfulness_4'] = clean_data['mindfulness_4'].fillna(1)
clean_data['mindfulness_5'] = clean_data['mindfulness_5'].fillna(1)
clean_data['mindfulness_6'] = clean_data['mindfulness_6'].fillna(1)
clean_data['mindfulness_7'] = clean_data['mindfulness_7'].fillna(1)
clean_data['mindfulness_8'] = clean_data['mindfulness_8'].fillna(1)
```

In [63]:

```
clean_data['witness_gun'] = data['witness_gun'] - 1
clean_data['witness_gun_gang'] = data['witness_gun_gang'] - 1
clean_data['perp_assltgun'] = data['perp_assltgun'] - 1
```

```
clean_data['perp_gun_gang'] = data['perp_gun_gang'] - 1
clean_data['vict_ass_gun'] = data['vict_ass_gun'] - 1
clean_data['vict_ass_gun_gang'] = data['vict_ass_gun_gang'] - 1
clean_data['vict_ass_gun_inj'] = data['vict_ass_gun_inj'] - 1
clean_data['avoidpolice'] = data['avoidpolice'] - 1
clean_data['gunaccess'] = data['gunaccess'] - 1
clean_data['gang_cur'] = data['gang_cur'] - 1
clean_data['gang_frmr'] = data['gang_frmr'] - 1
```

In [64]:

```
clean_data['witness_gun'] = clean_data['witness_gun'].fillna(0)
clean_data['witness_gun_gang'] = clean_data['witness_gun_gang'].fillna(0)
clean_data['perp_assltgun'] = clean_data['perp_assltgun'].fillna(0)
clean_data['perp_gun_gang'] = clean_data['perp_gun_gang'].fillna(0)
clean_data['vict_ass_gun'] = clean_data['vict_ass_gun'].fillna(0)
clean_data['vict_ass_gun_gang'] = clean_data['vict_ass_gun_gang'].fillna(0)
clean_data['vict_ass_gun_inj'] = clean_data['vict_ass_gun_inj'].fillna(0)
clean_data['avoidpolice'] = clean_data['avoidpolice'].fillna(0)
clean_data['gunaccess'] = clean_data['gunaccess'].fillna(0)
clean_data['gunaccess'] = clean_data['gunaccess'].replace(2, 1)
clean_data['gang_cur'] = clean_data['gang_cur'].fillna(0)
clean_data['gang_frmr'] = clean_data['gang_frmr'].fillna(0)
```

In [65]:

```
clean_data['gang_age'] = data['gang_age_1'].fillna(30)
```

In [66]:

```
clean_data['gang_provide_1'] = data['gang_provide_1'].fillna(0)
clean_data['gang_provide_2'] = data['gang_provide_2'].fillna(0)
clean_data['gang_provide_3'] = data['gang_provide_3'].fillna(0)
clean_data['gang_provide_4'] = data['gang_provide_4'].fillna(0)
clean_data['gang_provide_5'] = data['gang_provide_5'].fillna(0)
clean_data['gang_provide_6'] = data['gang_provide_6'].fillna(0)
clean_data['gang_provide_7'] = data['gang_provide_7'].fillna(0)
clean_data['gang_provide_8'] = data['gang_provide_8'].fillna(0)
clean_data['gang_provide_9'] = data['gang_provide_9'].fillna(0)
```

In [67]:

```
same = []

for i in data['gang_race']:
    if i == 1:
        same_ = 1
    else:
        same_ = 0
    same.append(same_)

clean_data['gang_same_race'] = same

#did you join a gang of the same race
```

In [68]:

```
male = []
female = []

for i in data['gang_gender']:
    if i == 1:
        male_ = 1
        female_ = 0
    elif i == 2:
        male_ = 0
        female_ = 0
    else:
        male_ = 0
        female_ = 0
    male.append(male_)
    female.append(female_)

clean_data['gang_mostly_male'] = male
clean_data['gang_mostly_female'] = female

# did you join a gang of the same gender, and we're creating 2 columns to show yes or no
for male and female
```

In [69]:

```
clean_data['gang_aff_1'] = data['gang_aff_1'].fillna(0)
clean_data['gang_aff_2'] = data['gang_aff_2'].fillna(0)
clean_data['gang_aff_3'] = data['gang_aff_3'].fillna(0)
clean_data['gang_aff_4'] = data['gang_aff_4'].fillna(0)
clean_data['gang_aff_5'] = data['gang_aff_5'].fillna(0)
clean_data['gang_aff_6'] = data['gang_aff_6'].fillna(0)
clean_data['gang_aff_7'] = data['gang_aff_7'].fillna(0)
clean_data['gang_aff_8'] = data['gang_aff_8'].fillna(0)
clean_data['gang_aff_9'] = data['gang_aff_9'].fillna(0)
clean_data['gang_aff_10'] = data['gang_aff_10'].fillna(0)
```

In [70]:

```
enc = []

for i in data['gang_enc']:
    if i == 2:
        enc_ = 1
    else:
        enc_ = 0
    enc.append(enc_)

clean_data['gang_enc'] = enc
# would you encourage others to join a gang
```

In [71]:

```
clean_data['juggalo'] = data['juggalo'] - 1
clean_data['streetfamily'] = data['streetfamily'] - 1
clean_data['juggalo'] = clean_data['juggalo'].fillna(0)
clean_data['streetfamily'] = clean_data['streetfamily'].fillna(0)
```

In [72]:

```
clean_data['mh_depress_1'] = data['mh_depress_1'].fillna(1)
clean_data['mh_depress_2'] = data['mh_depress_2'].fillna(1)
clean_data['mh_depress_3'] = data['mh_depress_3'].fillna(1)
clean_data['mh_depress_4'] = data['mh_depress_4'].fillna(1)
clean_data['mh_depress_5'] = data['mh_depress_5'].fillna(1)
clean_data['mh_depress_6'] = data['mh_depress_6'].fillna(1)
clean_data['mh_depress_7'] = data['mh_depress_7'].fillna(1)
clean_data['mh_depress_8'] = data['mh_depress_8'].fillna(1)
clean_data['mh_depress_9'] = data['mh_depress_9'].fillna(1)
```

In [73]:

```
clean_data['mh_mult_1'] = data['mh_mult_1'].fillna(1)
clean_data['mh_mult_2'] = data['mh_mult_2'].fillna(1)
clean_data['mh_mult_3'] = data['mh_mult_3'].fillna(1)
clean_data['mh_mult_5'] = data['mh_mult_5'].fillna(1)
clean_data['mh_mult_6'] = data['mh_mult_6'].fillna(1)
clean_data['mh_mult_7'] = data['mh_mult_7'].fillna(1)
```

In [74]:

```
clean_data['ptsd_1_2'] = data['ptsd_1_2'] - 1
clean_data['ptsd_1_3'] = data['ptsd_1_3'] - 1
clean_data['ptsd_1_4'] = data['ptsd_1_4'] - 1
clean_data['ptsd_1_5'] = data['ptsd_1_5'] - 1
```

In [75]:

In [75]:

```
clean_data['ptsd_1_2'] = clean_data['ptsd_1_2'].fillna(0)
clean_data['ptsd_1_3'] = clean_data['ptsd_1_3'].fillna(0)
clean_data['ptsd_1_4'] = clean_data['ptsd_1_4'].fillna(0)
clean_data['ptsd_1_5'] = clean_data['ptsd_1_5'].fillna(0)
```

In [76]:

```
clean_data['adhd_dx_2'] = data['adhd_dx_2'] - 1
clean_data['adhd_dx_3'] = data['adhd_dx_3'] - 1
clean_data['adhd_dx_4'] = data['adhd_dx_4'] - 1
clean_data['adhd_dx_5'] = data['adhd_dx_5'] - 1
clean_data['adhd_dx_6'] = data['adhd_dx_6'] - 1
clean_data['adhd_dx_7'] = data['adhd_dx_7'] - 1
```

In [77]:

```
clean_data['adhd_dx_2'] = clean_data['adhd_dx_2'].fillna(0)
clean_data['adhd_dx_3'] = clean_data['adhd_dx_3'].fillna(0)
clean_data['adhd_dx_4'] = clean_data['adhd_dx_4'].fillna(0)
clean_data['adhd_dx_5'] = clean_data['adhd_dx_5'].fillna(0)
clean_data['adhd_dx_6'] = clean_data['adhd_dx_6'].fillna(0)
clean_data['adhd_dx_7'] = clean_data['adhd_dx_7'].fillna(0)
```

In [78]:

```
clean_data['mh_current'] = data['mh_current'].replace(2,0)
clean_data['mh_current'] = clean_data['mh_current'].replace(3,0)
clean_data['mh_current'] = clean_data['mh_current'].fillna(0)
```

In [79]:

```
clean_data['mh_overall_1'] = data['mh_overall_1'].fillna(0)
clean_data['mh_overall_2'] = data['mh_overall_2'].fillna(0)
clean_data['mh_overall_3'] = data['mh_overall_3'].fillna(0)
clean_data['mh_overall_4'] = data['mh_overall_4'].fillna(0)
clean_data['mh_overall_5'] = data['mh_overall_5'].fillna(0)
clean_data['mh_overall_6'] = data['mh_overall_6'].fillna(0)
clean_data['mh_overall_7'] = data['mh_overall_7'].fillna(0)
```

In [80]:

```
clean_data['suic_thought'] = data['suic_thought'] - 1
clean_data['suic_attempt'] = data['suic_attempt'] - 1
```

In [81]:

```
clean_data['suic_thought'] = clean_data['suic_thought'].fillna(0)
clean_data['suic_attempt'] = clean_data['suic_attempt'].fillna(0)
```

In [82]:

```
clean_data['perc_stress1'] = data['perc_stress1'].fillna(1)
clean_data['perc_stress2'] = data['perc_stress2'].fillna(1)
clean_data['perc_stress3'] = data['perc_stress3'].fillna(1)
clean_data['perc_stress4'] = data['perc_stress4'].fillna(1)
```

In [83]:

```
clean_data['med_ever'] = data['med_ever'] - 1
clean_data['med_12'] = data['med_12'] - 1
clean_data['ther_ever'] = data['ther_ever'] - 1
clean_data['ther_12'] = data['ther_12'] - 1
clean_data['er_ever'] = data['er_ever'] - 1
clean_data['er_12'] = data['er_12'] - 1
clean_data['hospit_ever'] = data['hospit_ever'] - 1
clean_data['hospit_12'] = data['hospit_12'] - 1
clean_data['unmet_ever'] = data['unmet_ever'] - 1
clean_data['unmet_12'] = data['unmet_12'] - 1
```

In [84]:

```
clean_data['med_ever'] = clean_data['med_ever'].fillna(0)
clean_data['med_12'] = clean_data['med_12'].fillna(0)
clean_data['ther_ever'] = clean_data['ther_ever'].fillna(0)
clean_data['ther_12'] = clean_data['ther_12'].fillna(0)
clean_data['er_ever'] = clean_data['er_ever'].fillna(0)
clean_data['er_12'] = clean_data['er_12'].fillna(0)
clean_data['hospit_ever'] = clean_data['hospit_ever'].fillna(0)
clean_data['hospit_12'] = clean_data['hospit_12'].fillna(0)
clean_data['unmet_ever'] = clean_data['unmet_ever'].fillna(0)
clean_data['unmet_12'] = clean_data['unmet_12'].fillna(0)
```

In [85]:

```
mh_perceive = []

for i in data['mhneed_perceive']:
    if i == 1 or i == 3:
        mh_ = 1
    else:
        mh_ = 0
    mh_perceive.append(mh_)

clean_data['mhneed_perceive'] = mh_perceive
# do you tihnk you need mental health treatment
```

In [86]:

```
clean_data['helpseek_scale_1'] = data['helpseek_scale_1'].fillna(1)
clean_data['helpseek_scale_2'] = data['helpseek_scale_2'].fillna(1)
clean_data['helpseek_scale_3'] = data['helpseek_scale_3'].fillna(1)
clean_data['helpseek_scale_4'] = data['helpseek_scale_4'].fillna(1)
clean_data['helpseek_scale_5'] = data['helpseek_scale_5'].fillna(1)
clean_data['helpseek_scale_6'] = data['helpseek_scale_6'].fillna(1)
clean_data['helpseek_scale_7'] = data['helpseek_scale_7'].fillna(1)
clean_data['helpseek_scale_8'] = data['helpseek_scale_8'].fillna(1)
clean_data['desirehelp_1'] = data['desirehelp_1'].fillna(1)
clean_data['smoke_2'] = data['smoke_2'].fillna(1)
clean_data['alc_30'] = data['alc_30'].fillna(1)
```

In [87]:

```
none = []
rarely = []
weekly = []
regularly = []

none_ = 0
rarely_ = 0
weekly_ = 0
regularly_ = 0

for i in data['binge_30']:
    if i == 1:
        none_ = 1
    elif i == 2 or i == 3:
        rarely_ = 1
    elif i == 4:
        weekly_ = 1
    elif i == 5 or i == 6 or i == 7:
        regularly_ = 1
    else:
        none_ = 1

    none.append(none_)
    rarely.append(rarely_)
    weekly.append(weekly_)
    regularly.append(regularly_)
```

```
clean_data['binge_none'] = none
clean_data['binge_rarely'] = rarely
clean_data['binge_weekly'] = weekly
clean_data['binge_regularly'] = regularly

#how regularly do you binge if at all? do you not binge, rarely binge, etc.
```

In [88]:

```
none = []
rarely = []
weekly = []
regularly = []

none_ = 0
rarely_ = 0
weekly_ = 0
regularly_ = 0

for i in data['marj_30']:
    if i == 1:
        none_ = 1
    elif i == 2:
        rarely_ = 1
    elif i == 3:
        weekly_ = 1
    elif i == 4 or i ==  5 or i == 6:
        regularly_ = 1
    else:
        none_ = 1

    none.append(none_)
    rarely.append(rarely_)
    weekly.append(weekly_)
    regularly.append(regularly_)


clean_data['marj_none'] = none
clean_data['marj_rarely'] = rarely
clean_data['marj_weekly'] = weekly
clean_data['marj_regularly'] = regularly
# have you used weed in the last 30 days if so how many times and we're making them a dummy variable
```

In [89]:

```
clean_data['marj_access_1'] = data['marj_access_1'].fillna(0)
clean_data['marj_access_2'] = data['marj_access_2'].fillna(0)
clean_data['marj_access_3'] = data['marj_access_3'].fillna(0)
clean_data['marj_access_4'] = data['marj_access_4'].fillna(0)
clean_data['marj_access_5'] = data['marj_access_5'].fillna(0)
```

In [90]:

```
usedmore = []

for i in data['mhneed_perceive']:
    if i == 2:
        usedmore_ = 1
    else:
        usedmore_ = 0
    usedmore.append(usedmore_)

clean_data['marj_usedmore'] = usedmore
#do you think you've used more weed since your  mental health thing
```

In [91]:

```
policy = []

for i in data['marj_policy']:
    if i == 2:
        policy_ = 1
    else:
        policy_ = 0
    policy.append(policy_)

clean_data['marj_policy'] = policy

#did weed laws dictate which city you're currently in
```

In [92]:

```
none = []
rarely = []
weekly = []
regularly = []

none_ = 0
rarely_ = 0
weekly_ = 0
regularly_ = 0

for i in data['rx_30']:
    if i == 1:
        none_ = 1
    elif i == 2:
        rarely_ = 1
    elif i == 3:
        weekly_ = 1
    elif i == 4 or i ==  5 or i == 6:
        regularly_ = 1
    else:
        none_ = 1

    none.append(none_)
    rarely.append(rarely_)
    weekly.append(weekly_)
    regularly.append(regularly_)


clean_data['rx_none'] = none
clean_data['rx_rarely'] = rarely
clean_data['rx_weekly'] = weekly
clean_data['rx_regularly'] = regularly
#have you taken prescription drugs recently, if so how much and how often
```

In [93]:

```
clean_data['rx_type_30_1'] = data['rx_type_30_1'].fillna(0)
clean_data['rx_type_30_2'] = data['rx_type_30_2'].fillna(0)
clean_data['rx_type_30_3'] = data['rx_type_30_3'].fillna(0)
clean_data['rx_type_30_4'] = data['rx_type_30_4'].fillna(0)
```

In [94]:

```
clean_data['rx_how_30_1'] = data['rx_how_30_1'].fillna(0)
clean_data['rx_how_30_2'] = data['rx_how_30_2'].fillna(0)
clean_data['rx_how_30_3'] = data['rx_how_30_3'].fillna(0)
clean_data['rx_how_30_4'] = data['rx_how_30_4'].fillna(0)
clean_data['rx_how_30_5'] = data['rx_how_30_5'].fillna(0)
clean_data['rx_how_30_6'] = data['rx_how_30_6'].fillna(0)
clean_data['rx_how_30_7'] = data['rx_how_30_7'].fillna(0)
clean_data['rx_how_30_8'] = data['rx_how_30_8'].fillna(0)
clean_data['rx_how_30_9'] = data['rx_how_30_9'].fillna(0)
```

In [95]:

```
inject = []

for i in data['inject_30']:
    if i == 1:
        inject_ = 1
    else:
        inject_ = 0
    inject.append(inject_)

clean_data['inject_30'] = inject
# have you taken drugs that you inject recently, if so how often. If you have then 1 if not then 0
```

In [96]:

```
clean_data['needle_share_30'] = data['needle_share_30'] - 1
clean_data['sub_treat'] = data['sub_treat'] - 1
clean_data['subtreat_pastyear'] = data['subtreat_pastyear'] - 1
clean_data['cage1'] = data['cage1'] - 1
clean_data['cage2'] = data['cage2'] - 1
clean_data['cage3'] = data['cage3'] - 1
clean_data['cage4'] = data['cage4'] - 1
```

In [97]:

```
clean_data['needle_share_30'] = clean_data['needle_share_30'].fillna(0)
clean_data['sub_treat'] = clean_data['sub_treat'].fillna(0)
clean_data['subtreat_pastyear'] = clean_data['subtreat_pastyear'].fillna(0)
clean_data['cage1'] = clean_data['cage1'].fillna(0)
clean_data['cage2'] = clean_data['cage2'].fillna(0)
clean_data['cage3'] = clean_data['cage3'].fillna(0)
clean_data['cage4'] = clean_data['cage4'].fillna(0)
```

In [98]:

```
clean_data['techaccess_1'] = data['techaccess_1'].fillna(0)
clean_data['techaccess_2'] = data['techaccess_2'].fillna(0)
clean_data['techaccess_3'] = data['techaccess_3'].fillna(0)
clean_data['techaccess_4'] = data['techaccess_4'].fillna(0)
clean_data['techaccess_5'] = data['techaccess_5'].fillna(0)
```

In [99]:

```
clean_data['socmeduse_1'] = data['socmeduse_1'].fillna(0)
clean_data['socmeduse_2'] = data['socmeduse_2'].fillna(0)
clean_data['socmeduse_3'] = data['socmeduse_3'].fillna(0)
clean_data['socmeduse_4'] = data['socmeduse_4'].fillna(0)
clean_data['socmeduse_5'] = data['socmeduse_5'].fillna(0)
clean_data['socmeduse_6'] = data['socmeduse_6'].fillna(0)
clean_data['socmeduse_7'] = data['socmeduse_7'].fillna(0)
clean_data['socmeduse_8'] = data['socmeduse_8'].fillna(0)
```

In [100]:

```
clean_data['socmed_connect_1'] = data['socmed_connect_1'] - 1
clean_data['socmed_connect_2'] = data['socmed_connect_2'] - 1
```

In [101]:

```
clean_data['socmed_connect_1'] = clean_data['socmed_connect_1'].fillna(0)
clean_data['socmed_connect_2'] = clean_data['socmed_connect_2'].fillna(0)
```

In [102]:

```
none = []
rarely = []
daily = []
often = []

none_ = 0
rarely_ = 0
daily_ = 0
often_ = 0
```

```
for i in data['socmedtime']:
    if i == 6:
        none_ = 1
    elif i == 3 or i == 4 or i == 5:
        rarely_ = 1
    elif i == 2:
        daily_ = 1
    elif i == 1:
        often_ = 1
    else:
        none_ = 1

    none.append(none_)
    rarely.append(rarely_)
    daily.append(daily_)
    often.append(often_)


clean_data['socmed_none'] = none
clean_data['socmed_rarely'] = rarely
clean_data['socmed_daily'] = daily
clean_data['socmed_often'] = often
# how often do you use social media
```

In [103]:

```
clean_data['infoonline_1'] = data['infoonline_1'].fillna(0)
clean_data['infoonline_2'] = data['infoonline_2'].fillna(0)
clean_data['infoonline_3'] = data['infoonline_3'].fillna(0)
clean_data['infoonline_4'] = data['infoonline_4'].fillna(0)
clean_data['infoonline_5'] = data['infoonline_5'].fillna(0)
clean_data['infoonline_6'] = data['infoonline_6'].fillna(0)
clean_data['infoonline_7'] = data['infoonline_7'].fillna(0)
clean_data['infoonline_8'] = data['infoonline_8'].fillna(0)
clean_data['infoonline_9'] = data['infoonline_9'].fillna(0)
clean_data['infoonline_10'] = data['infoonline_10'].fillna(0)
clean_data['infoonline_11'] = data['infoonline_11'].fillna(0)
clean_data['infoonline_12'] = data['infoonline_12'].fillna(0)
```

In [104]:

```
clean_data['socservonline_1'] = data['socservonline_1'].fillna(0)
clean_data['socservonline_2'] = data['socservonline_2'].fillna(0)
clean_data['socservonline_3'] = data['socservonline_3'].fillna(0)
clean_data['socservonline_4'] = data['socservonline_4'].fillna(0)
clean_data['socservonline_5'] = data['socservonline_5'].fillna(0)
clean_data['socservonline_6'] = data['socservonline_6'].fillna(0)
clean_data['socservonline_7'] = data['socservonline_7'].fillna(0)
clean_data['socservonline_8'] = data['socservonline_8'].fillna(0)
clean_data['socservonline_9'] = data['socservonline_9'].fillna(0)
clean_data['socservonline_10'] = data['socservonline_10'].fillna(0)
clean_data['socservonline_12'] = data['socservonline_12'].fillna(0)
clean_data['socservonline_14'] = data['socservonline_14'].fillna(0)
```

In [105]:

```
none = []
one = []
two = []
three = []
four = []
five = []
sixplus = []

none_ = 0
one_ = 0
two_ = 0
three_ = 0
four_ = 0
five_ = 0
sixplus_ = 0
```

```
for i in data['life_sexpartners']:
    if i == 4:
        one_ = 1
    elif i == 5:
        two_ = 1
    elif i == 6:
        three_ = 1
    elif i == 7:
        four_ = 1
    elif i == 8:
        five_ = 1
    elif i == 9:
        sixplus_ = 1
    else:
        none_ = 1

    none.append(none_)
    one.append(one_)
    two.append(two_)
    three.append(three_)
    four.append(four_)
    five.append(five_)
    sixplus.append(sixplus_)


clean_data['life_sexpartners_none'] = none
clean_data['life_sexpartners_one'] = one
clean_data['life_sexpartners_two'] = two
clean_data['life_sexpartners_three'] = three
clean_data['life_sexpartners_four'] = four
clean_data['life_sexpartners_five'] = five
clean_data['life_sexpartners_sixplus'] = sixplus

# how many life or sex partners have you had recently
```

In [106]:

```
clean_data['lastsextype_4'] = data['lastsextype_4'].fillna(0)
clean_data['lastsextype_5'] = data['lastsextype_5'].fillna(0)
clean_data['lastsextype_6'] = data['lastsextype_6'].fillna(0)
clean_data['lastsextype_7'] = data['lastsextype_7'].fillna(0)
clean_data['lastsextype_8'] = data['lastsextype_8'].fillna(0)
clean_data['lastsextype_9'] = data['lastsextype_9'].fillna(0)

clean_data['look_sexpart_4'] = data['look_sexpart_4'].fillna(0)
clean_data['look_sexpart_5'] = data['look_sexpart_5'].fillna(0)
clean_data['look_sexpart_6'] = data['look_sexpart_6'].fillna(0)
clean_data['look_sexpart_7'] = data['look_sexpart_7'].fillna(0)
clean_data['look_sexpart_8'] = data['look_sexpart_8'].fillna(0)
clean_data['look_sexpart_9'] = data['look_sexpart_9'].fillna(0)
clean_data['look_sexpart_10'] = data['look_sexpart_10'].fillna(0)
clean_data['look_sexpart_11'] = data['look_sexpart_11'].fillna(0)
clean_data['look_sexpart_12'] = data['look_sexpart_12'].fillna(0)
clean_data['look_sexpart_14'] = data['look_sexpart_14'].fillna(0)

clean_data['jugg_provide_1'] = data['jugg_provide_1'].fillna(0)
clean_data['jugg_provide_2'] = data['jugg_provide_2'].fillna(0)
clean_data['jugg_provide_3'] = data['jugg_provide_3'].fillna(0)
clean_data['jugg_provide_4'] = data['jugg_provide_4'].fillna(0)
clean_data['jugg_provide_5'] = data['jugg_provide_5'].fillna(0)
clean_data['jugg_provide_6'] = data['jugg_provide_6'].fillna(0)
clean_data['jugg_provide_7'] = data['jugg_provide_7'].fillna(0)
clean_data['jugg_provide_8'] = data['jugg_provide_8'].fillna(0)
clean_data['jugg_provide_9'] = data['jugg_provide_9'].fillna(0)
```

In [107]:

```
same = []

for i in data['jugg_race']:
    if i == 1:
        same_ = 1
    else:
        same_ = 0
    same.append(same_)

clean_data['jugg_same_race'] = same
# asking about juggalo lifestyles?
```

In [108]:

```
male = []
female = []

for i in data['jugg_gen']:
    if i == 1:
        male_ = 1
        female_ = 0
    elif i == 2:
        male_ = 0
        female_ = 0
    else:
        male_ = 0
        female_ = 0
    male.append(male_)
    female.append(female_)

clean_data['jugg_mostly_male'] = male
clean_data['jugg_mostly_female'] = female
#gender of the juggalos you hung out with
```

In [109]:

```
enc = []

for i in data['jugg_enc']:
    if i == 2:
        enc_ = 1
    else:
        enc_ = 0
    enc.append(enc_)

clean_data['jugg_enc'] = enc
# woudl you encourage people to be juggalos
```

In [110]:

```
clean_data['ego_heroin'] = data['ego_heroin'].fillna(0)
clean_data['ego_cocaine'] = data['ego_cocaine'].fillna(0)
clean_data['ego_crack'] = data['ego_crack'].fillna(0)
clean_data['ego_spice'] = data['ego_spice'].fillna(0)
clean_data['ego_ecstasy'] = data['ego_ecstasy'].fillna(0)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [111]:

```
data['druguse_30_1'].isna().sum()
```

Out[111]:

```
145
```

In [112]:

```
data['druguse_30_2'].isna().sum()
```

Out[112]:

```
145
```

In [113]:

```
data['druguse_30_3'].isna().sum()
```

Out[113]:

```
160
```

In [114]:

```
data['druguse_30_4'].isna().sum()
```

Out[114]:

```
145
```

In [115]:

```
data['druguse_30_5'].isna().sum()
```

Out[115]:

```
155
```

In [116]:

```
data['druguse_30_6'].isna().sum()
```

Out[116]:

```
140
```

In [ ]:

```

```

In [ ]:

```

```

here we're defining our testing values that we're looking for. afterwards we're going to be dropping them from our data and then creating a new dataframe with all of these values

In [117]:

```
cocaine = []

for i in data['druguse_30_1']:
    if i == 2 or i == 3 or i == 4 or i == 5 or i ==6:
        cocaine_ = 1
    else:
        cocaine_ = 0
    cocaine.append(cocaine_)

clean_data['cocaine_user'] = cocaine
```

In [118]:

```
crack = []

for i in data['druguse_30_2']:
    if i == 2 or i == 3 or i == 4 or i == 5 or i ==6:
        crack_ = 1
    else:
        crack_ = 0
    crack.append(crack_)

clean_data['crack_user'] = crack
```

In [119]:

```
heroin = []

for i in data['druguse_30_3']:
    if i == 2 or i == 3 or i == 4 or i == 5 or i ==6:
        heroin_ = 1
    else:
        heroin_ = 0
    heroin.append(heroin_)

clean_data['heroin_user'] = heroin
```

In [120]:

```
meth = []

for i in data['druguse_30_4']:
    if i == 2 or i == 3 or i == 4 or i == 5 or i ==6:
        meth_ = 1
    else:
        meth_ = 0
    meth.append(meth_)

clean_data['meth_user'] = meth
```

In [121]:

```
ecstasy = []

for i in data['druguse_30_5']:
    if i == 2 or i == 3 or i == 4 or i == 5 or i ==6:
        ecstasy_ = 1
    else:
        ecstasy_ = 0
    ecstasy.append(ecstasy_)

clean_data['ecstasy_user'] = ecstasy
```

In [122]:

```
spice = []

for i in data['druguse_30_6']:
    if i == 2 or i == 3 or i == 4 or i == 5 or i ==6:
        spice_ = 1
    else:
        spice_ = 0
    spice.append(spice_)

clean_data['spice_user'] = spice
```

In [ ]:

```

```

In [ ]:

In [ ]:

```

```

In [ ]:

```

```

In [123]:

```
clean_data.head(50)
```

Out[123]:

	pid	screen1_sleep_shelter	screen1_sleep_insecure	screen1_sleep_secure	screen3_age	realm_score_help_needed	gende
10	1006	0	1	0	22.0	0	
11	1006	0	1	0	22.0	0	
12	1006	0	1	0	22.0	0	
13	1006	0	1	0	22.0	0	
14	1006	0	1	0	22.0	0	
15	1007	0	1	0	24.0	0	
16	1007	0	1	0	24.0	0	
17	1007	0	1	0	24.0	0	
18	1007	0	1	0	24.0	0	
19	1007	0	1	0	24.0	0	
20	1008	1	1	0	19.0	0	
21	1008	1	1	0	19.0	0	
22	1008	1	1	0	19.0	0	
23	1008	1	1	0	19.0	0	
24	1008	1	1	0	19.0	0	
25	1009	1	1	0	19.0	0	
26	1009	1	1	0	19.0	0	
27	1009	1	1	0	19.0	0	
28	1009	1	1	0	19.0	0	
29	1009	1	1	0	19.0	0	
35	1011	1	1	0	19.0	0	
36	1011	1	1	0	19.0	0	
37	1011	1	1	0	19.0	0	
38	1011	1	1	0	19.0	0	
39	1011	1	1	0	19.0	0	
40	1012	1	1	0	23.0	0	
41	1012	1	1	0	23.0	0	
42	1012	1	1	0	23.0	0	
43	1012	1	1	0	23.0	0	
44	1012	1	1	0	23.0	0	
45	1013	1	1	0	18.0	0	
46	1013	1	1	0	18.0	0	
47	1013	1	1	0	18.0	0	
48	1013	1	1	0	18.0	0	
49	1013	1	1	0	18.0	0	

	pid	screen1_sleep_shelter	screen1_sleep_insecure	screen1_sleep_secure	screen3_age	realm_score_help_needed	gende
50	1014	1	1	0	22.0	0	
51	1014	1	1	0	22.0	0	
52	1014	1	1	0	22.0	0	
53	1014	1	1	0	22.0	0	
54	1014	1	1	0	22.0	0	
55	1015	1	1	0	19.0	0	
56	1015	1	1	0	19.0	0	
57	1015	1	1	0	19.0	0	
58	1015	1	1	0	19.0	0	
59	1015	1	1	0	19.0	0	
60	1016	1	1	0	20.0	0	
61	1016	1	1	0	20.0	0	
62	1016	1	1	0	20.0	0	
63	1016	1	1	0	20.0	0	
64	1016	1	1	0	20.0	0	

50 rows × 358 columns

In [ ]:

In [124]:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score

# Here we're dropping the values that we're looking to predict from our new dataset that'
s all cleaned up

clean_data2 = clean_data.drop('cocaine_user', axis =1)

clean_data2 = clean_data.drop('crack_user', axis =1)

clean_data2 = clean_data.drop('heroin_user', axis =1)

clean_data2 = clean_data.drop('meth_user', axis =1)

clean_data2 = clean_data.drop('ecstasy_user', axis =1)

clean_data2 = clean_data.drop('spice_user', axis =1)


X = clean_data2
y = clean_data['cocaine_user']
y2 = clean_data['crack_user']
y3 = clean_data['heroin_user']
y4 = clean_data['meth_user']

y5 = clean_data['ecstasy_user']

y6 = clean_data['spice_user']


# next we're initializiing our dataset
clf = RandomForestClassifier(max_depth=2, random_state=0)
```

In [125]:

```
clf.fit(X, y)
```

Out[125]:

```
RandomForestClassifier(max_depth=2, random_state=0)
```

We used this next section to better understand the relationships between each of our prediction values relative to our model. However, we want to predict the values all together so this wasn't necessarily the best idea.

In [126]:

```
crossScores = cross_val_score(clf, X, y, cv=10, scoring="roc_auc")
crossScores2 = cross_val_score(clf, X, y2, cv=10, scoring="roc_auc")
crossScores3 = cross_val_score(clf, X, y3, cv=10, scoring="roc_auc")
crossScores4 = cross_val_score(clf, X, y4, cv=10, scoring="roc_auc")
crossScores5 = cross_val_score(clf, X, y5, cv=10, scoring="roc_auc")
crossScores6 = cross_val_score(clf, X, y6, cv=10, scoring="roc_auc")
```

In [127]:

```
print("Cocaine ", np.mean(crossScores))
print("Crack ", np.mean(crossScores))
print("Heroin ", np.mean(crossScores))
print("Meth ", np.mean(crossScores))
print("Ecstasy ", np.mean(crossScores))
print("Spice ", np.mean(crossScores))
```

```
Cocaine  0.9913766378842664
Crack  0.9913766378842664
Heroin  0.9913766378842664
Meth  0.9913766378842664
Ecstasy  0.9913766378842664
Spice  0.9913766378842664
```

In [128]:

```
clean_data.shape
```

Out[128]:

```
(4785, 358)
```

In [130]:

```
frame = {'cocaine':y, 'crack':y2, 'heroin':y3, 'meth':y4, 'ecstasy':y5, 'spice':y6}
Y = pd.DataFrame(frame) # this is our y dataframe
```

In [131]:

```
Y
```

Out[131]:

	0	1	2	3	4	5
0	10 1 11 1 12 1 13 1 14 ...	10 1 11 1 12 1 13 1 14 ...	10 1 11 1 12 1 13 1 14 ...	10 1 11 1 12 1 13 1 14 ...	10 1 11 1 12 1 13 1 14 ...	10 1 11 1 12 1 13 1 14 ...

In [132]:

```
from sklearn.model_selection import train_test_split

# we want to do a train test split so we can better show some accuracy metrics
X_train, X_test, y_train, y_test = train_test_split(clean_data2, Y, test_size = .33, random_state=42)
```

The first model we're going to try is our random forest

classifier. this didn't do as great as we thought because we're predicting on many different values now rather than the individual values. this had an AUC score of about .53 which was above the first baseline but we could do much better

In [133]:

```
clf.fit(X_train, y_train)
```

Out[133]:

```
RandomForestClassifier(max_depth=2, random_state=0)
```

In [134]:

```
predictedValues = clf.predict(X_test)
```

In [135]:

```
predictedValues
```

Out[135]:

```
array([[0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0],
       ...,
       [0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 0, 0]])
```

In [136]:

```
from sklearn.metrics import roc_auc_score
# this is our roc_auc score but this isn't a great representation of our final metric
roc_auc_score(y_test, predictedValues)
```

Out[136]:

```
0.536811907178984
```

The next model we're going to try using is a binary relevance model with gaussian NB. Binary relevance comes from the skmultilearn package which specializes in multi label predictions. our roc_auc score was much higher than the original .54 but this could later be improved upon. We didn't do much research into why this was giving us that value but we trusted it at face value. This is also not cross validated since our value was still below the second baseline

In [137]:

```
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import MultinomialNB



classifier = BinaryRelevance(GaussianNB())
classifier.fit(X_train, y_train)

predictedValues2 = classifier.predict(X_test)

roc_auc_score(y_test, predictedValues2.toarray())
```

Out[137]:

```
0.7346450766112634
```

So we knew that with the binary relevance surrounding our base model would be the best way to do this. Binary Relevance is simply just a built in ensemble method that predicts on multiple labels for a problem which is exactly what we wanted. We thought We could try a SVC as our base model. Even with some hyperparameter tuning this model would probably not be the best that we can get since the output was a 0.5 roc_auc_score. This was definitly not what we wanted

In [138]:

```
from sklearn.svm import SVC

classifier2 = BinaryRelevance(SVC())
classifier2.fit(X_train, y_train)

predictedValues3 = classifier2.predict(X_test)

roc_auc_score(y_test, predictedValues3.toarray())
```

Out[138]:

```
0.5
```

our next attempt was to use a classifierchain. This did slightly better because this model essentially chains together what we're trying to predict. We, for instance, predict on spice and then use that prediction to predict on heroin, and so forth until all our predictions are complete. With an AUC score of .718 this was worse than our gaussian NB with binary relevance. we could have experimented more with this but we wanted to exhaust all options.

In [139]:

```
from skmultilearn.problem_transform import ClassifierChain
classifier = ClassifierChain(GaussianNB())
classifier.fit(X_train, y_train)
predictions = classifier.predict(X_test)
roc_auc_score(y_test, predictions.toarray())
```

Out[139]:

```
0.7180957051922262
```

By sheer luck we were able to find that a random forest classifier paired with a binary relevance ensemble produced a resonable auc score to us. We saw that it gave us an auc score of about .8 when we first ran it. We thought to increase the max_depth. When increasing the max depth we saw that the auc score can go up to .99 however we were unsure if we were overfitting the model. to check this we performed 10 fold cross validation which seemed to confirm that we were able to get an

AUC score of 1.0 on the entire dataset but this still didn't seem right to us. Instead we settled on a maximum depth of 10 instead of 15 to prevent any possible overfitting

In [155]:

```
# Max_depth increases accuracy, 15 gives us a 1.0 and that shouldn't be overfitting.
classifier = BinaryRelevance(RandomForestClassifier(max_depth=10, random_state=0))
classifier.fit(X_train, y_train)


predictions = classifier.predict(X_test)
roc_auc_score(y_test, predictions.toarray())
```

Out[155]:

```
0.8869263889506804
```

In [156]:

```
#god we hate this type of matrix so much #
type(predictions)
```

Out[156]:

```
scipy.sparse.csc.csc_matrix
```

We're going to perform cross validation over here now. we tried StratifiedKFold but that did not seem to work very well. Overall we ended up just making our own cross validation function. We used the KFold function to create our folds and then we simply just went rhoguh and ran our predictions on each of those folds. Finally we appended our cross validation scores to a new array that we eventually took the mean of to find a cross validated score of .99.

In [157]:

```
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import make_scorer
# here we're going to perform cross validation on the entire test set. It ac
#from sklearn.cross_validation import StratifiedKFold
classifier.fit(X, Y)
Xnew = X
Ynew = Y

#cross_val_score(classifier, Xnew, Ynew, cv=10, scoring=make_scorer(roc_auc_score))
#kf = StratifiedKFold(Y, n_splits = 10, indices=True)
```

In [158]:

```
from sklearn.model_selection import KFold
kf = KFold(n_splits=10)
Xnew = X.to_numpy()
Ynew = Y.to_numpy()

cvs_array = []
for train_index, test_index in kf.split(Xnew):
    #print("TRAIN:", train_index, "TEST:", test_index)
    X_train, X_test = Xnew[train_index], Xnew[test_index]
    y_train, y_test = Ynew[train_index], Ynew[test_index]
    predictions = classifier.predict(X_test)
    score = roc_auc_score(y_test, predictions.toarray())
    cvs_array.append(score)
    print(score)
```

```
kf.get_n_splits(X)
```

```
0.9962121212121212
0.9910394265232975
0.9943310657596371
0.9975490196078431
0.9971264367816092
0.9895264116575593
0.9910714285714285
0.9814814814814815
0.9807692307692308
0.9910714285714285
```

Out[158]:

```
10
```

In [159]:

```
np.mean(cvs_array)
```

Out[159]:

```
0.9910178050935636
```

Analysis: There isn't a whole lot to analyze here since our analysis was pretty much our cross validation score.

Model Applicability: I would definitely use this model to predict the drug usages of the homeless. It was able to predict with an accuracy of .99 whether someone would use a combination of certain drugs. I think the use case of this could be to help provide relief to the homeless. If a government organization was able to make homeless peopel take this survey for some kind of monetary incentive then they could better figure out where to put certain resources. For example, if a certain area was found to have a high amount of drug users or people who are predicted to use drugs, then extra services could be deployed there such as a squad of EMS technicians who carry narcan to prevent overdoses. This is just one possible application of this model which predicts at a high level of accuracy.

In [ ]:

```

```