Examining and predicting substance use risk through its contributing factors

Team Members: Jose Cordova, Vishal Harihar Ganapathy Krishnan, Shikhar Shukla, Shravani Basanthpur, Pravalikareddy Arakoti, Pallavi Singh, and Ananth Sai Varma Pericherla.

Indiana University–Purdue University Indianapolis, Indianapolis, Indiana, USA
jfcordov@iu.edu, vganapat@iu.edu, shikshuk@iu.edu, sbasanth@iu.edu, parakoti@iu.edu, singpall@iu.edu, aperich@iu.edu

Abstract. This project seeks to examine which factors contribute to substance use and substance use frequency. This work aims to identify populations that are more vulnerable to substance use and substance use frequency to assist in creating data-based interventions. Using data analysis, we identify which populations are most vulnerable to using different substances. Our hope is to provide information for public health entities to direct interventions toward those populations that need it the most.

Keywords: substance use, substance use frequency, data analysis, public health

1 Project Scope

1.1 Introduction

In 2015, a scientific network created by the United Nations and the World Health Organization created several recommendations for policymakers to address substance abuse. Among these recommendations, the network advised addressing substance use as a public health issue and implementing evidence-based prevention programs that identify and address risk and protective factors using data to drive policy decisions (Volkow et al., 2017)[i].

In public health, factors that increase the probability of an event occurring are known as risk factors, while protective factors reduce the probability of said event occurring (NIDA, 2020)[ii]. For example, according to the Centers for Disease Control and Prevention (2022), risk factors for the substance can be, among other things, low academic achievement or mental health issues.

Multiple studies have used the data from the National Survey on Substance Use and Health (Center for Behavioral Health Statistics and Quality, 2021)[iii] to determine these factors. Rosner et al. 2021 paper attempt to identify these factors using the survey's 2021 data by examining the link of substance abuse among sexual minorities to inequalities and unmet needs for mental health treatment[iv].

Waddell uses the 2002-2019 data from this survey to examine the risk of suffering from Alcohol Use Disorder[v]. While Lin et al. (2016) use the 2013 version of this survey to find the factors contributing to medical and recreational cannabis use[vi].

1.2 Aim

We aim to compare how different factors predict the risk of substance use and frequency of multiple types of substances. Using data from the National Survey on Drug Use and Health (Center for Behavioral Health Statistics and Quality, 2021), we seek to identify which factors collected in this the risk of substance use as well as the frequency of substance use in the past year by building two different types of machine learning models.

Regression models, We hope to use our regression models to use the examined factors to predict a participant's substance use frequency.

- Null hypothesis: The examined factors did not predict the frequency of substance use in the past year.
- Alternative hypothesis: The examined factors predicted the frequency of substance use in the past year.

Classification models, We hope to use our model to successfully classify participants (vectors) based on their examined factors as having never or ever consumed a specific substance.

- Null hypothesis: The examined factors did not predict substance use.
- Alternative hypothesis: The examined factors predicted substance use.

1.3 Purpose

Through machine learning models, we seek to determine which factors influence substance use and the frequency of substance use to identify risk and protective factors. As previously mentioned, substance abuse is a public health issue, we hope that with our model, we contribute towards evidence-based, data-driven solutions which seek to alleviate this aforementioned issue by allowing leaders in public health to identify which populations need it the most. Past research that has used the NSDUH has identified high population density, higher levels of socioeconomic vulnerability (Rosner et al., 2021), poor levels of self-reported health (Lin et al., 2021), and co-use of other substances (Waddell, 2021) as risk factors for substance use. We hope to consolidate some of the findings in these papers using data analysis to identify which factors are associated with substance use/frequency of use. Through this project, we can help public health entities identify which populations are at higher risk of using and abusing (frequency of use) substances.

2 Methodology

2.1 Steps of the project

The main focus of our project is using the NSDUH (2021) data to identify which factors contribute to substance use and abuse (frequency of use). To conduct our analysis, we used Python (matplotlib, pandas, seaborn). Furthermore, we used Google Colaboratory to share our notebooks with each other and Tableau to create visualizations. Our project consisted of the following stages, more on each step will be elaborated on.

1. Data Extraction and Coding
2. Data Analysis
3. Machine Learning Models and Evaluation

2.2 Team Members and Responsibilities

Team Members Responsibilities			
Names	Background	Responsibilities	Contributions
Jose Cordova	BA in Economics	Project proposal, Exploratory data analysis, final report	Project proposal, exploratory data analysis, final report
Vishal Harihar Ganapathy Krishnan	BE in Electronics and Communication Engg	Implement Machine Learning Models, Model Evaluation	Implement machine learning models, model evaluation
Shikhar Shukla	Bachelor of Dental Surgery	Visualizations using Seaborn And Tableau	Visualizations using seaborn and Tableau, LaTex formatting of the report, literature review for finalizing the independent variables, research questions.
Pravalika reddy Arakoti	Bachelor of Dental Surgery	Implement and evaluate random forest machine learning model	Implement and evaluate random forest and SVM classification machine learning model, hypothesis testing
Pallavi Singh	Bachelor of Dental Surgery	Project draft and final proposal proofreading, literature review	Project draft and final proposal proofreading, final report

		for finalizing the independent variables, research questions, data visualization Python and Tableau, final report proofreading	proofreading, data visualization using Python and Tableau, hypothesis testing LaTex formatting of the report.
Ananth Sai Varma Pericherla	Bachelor of Dental Surgery	Project Proposal formatting, final report formatting, Data Visualization using Tableau and Seaborn	Data visualization using seaborn and Tableau, LaTex formatting of the report.
Shravani Basanthpur	Pharmacology	Project Proposal formatting, Model Visualizations	Implement machine learning models, model evaluation, hypothesis testing

2.3 Project Challenges

There were multiple challenges while working on the project. These challenges ranged from having difficulties finding data to analyze to other technical obstacles, among other things. The first challenge was finding a dataset to analyze. This took a lot of time because a lot of the proposals were not approved by the TA due to a multitude of reasons. Finally, we received approval to use the NSDUH data. The second major challenge occurred when we received feedback on our proposal draft. Our proposal draft received 6/10 points, so we had to address multiple aspects of our draft. To address them, we assisted with virtual office hours. Specifically, our difficulties were regarding how our research questions were tied to our deliverables. The third major challenge was accessing the data. The dataset weighs more than 250 Mb; even opening it in Excel can take a while. Initially, we thought the solution would be to connect to the data using PhpMyAdmin, by hosting said data. However, our teammate Vishal found the optimal solution: using pd.read_table() function, we were able to directly read the data from the website. This improved our workflow because we could share the code easily using Google collaboratory.

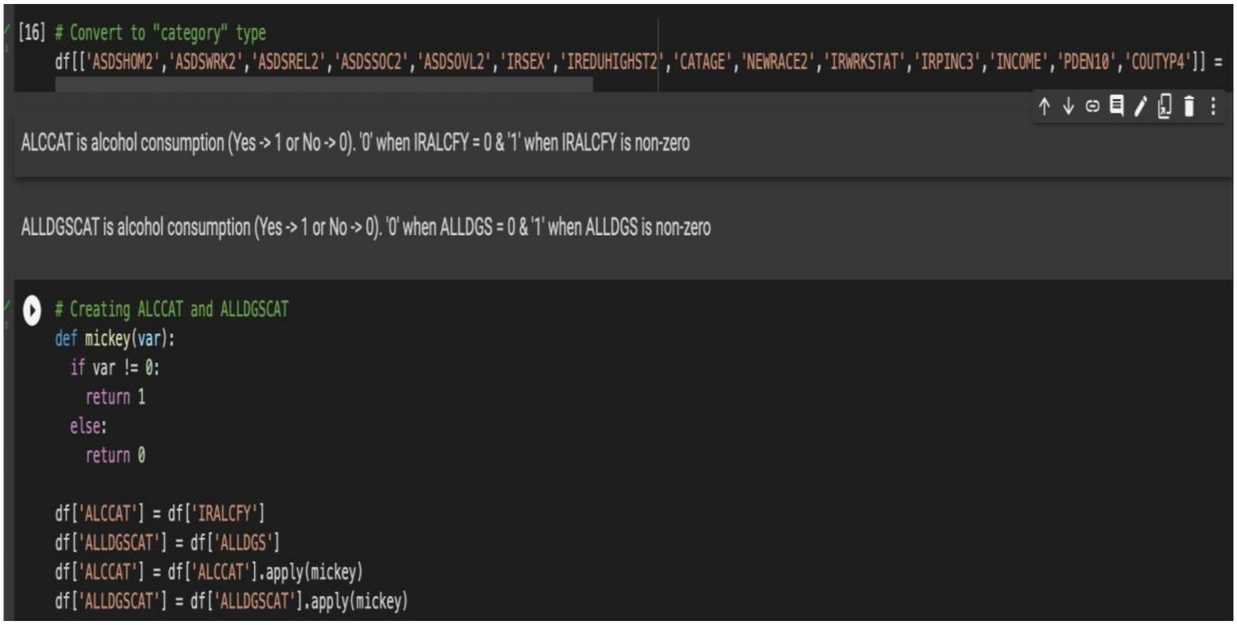
3 Data Extraction and Coding

The dataset was taken from the Substance Abuse and Mental Health Services Administration (SAMHSA) website. The National Survey on Substance Use and Health (Center for Behavioral Health Statistics and Quality, 2021), also known as the NSDUH. According to the dataset codebook (Center for Behavioral Health Statistics and Quality, 2021), the NSDUH contains items about first use, lifetime, annual, and month-to-date use of the following: Inhalants, tobacco, alcohol, marijuana, cocaine, hallucinogens, heroin, sedatives, pain relievers, tranquilizers, and stimulants. Additionally, the treatment history and perceived need for treatment of drug abuse are asked in the survey. Furthermore, gender, age, race, ethnicity, household composition, personal/family in- come, access to and coverage of health care, educational level, employment status, income level, veteran status, population density, and records of illegal activities or arrests are also included in the survey. The variables in the dataset can be qualitative, from nominal (for example, ethnicity or veteran status) to ordinal (for example, educational level), to quantitative (such as income level). The data can be downloaded in a TSV format on the NSDUH website. The survey contains over 2000 variables. By using previous literature, such as Lin et al. (2016), Waddell (2021), and Rosner et al. (2021), we narrowed down the variables we will include in our analysis. These variables are described in figure 1.

Categorical and Qualitative			
Nominal	Ordinal	Discrete	Continuous
Drug use (never/ever used)*, ethnicity, gender, health insurance, receiving government assistance, past year depression	Age group, employment, education, annual household income	Past year frequency of drug use*, level of disability from depression, perception of unmet need for mental health	Population density

*This includes the use of the following drugs: alcohol, marijuana, cocaine, crack, heroin, hallucinogens, inhalants, and meth.

Using the Pandas Python package, we accessed the desired variables directly from the SAMHSA's website. Furthermore, we combined the NSDUH datasets from 2015-2020 to have more data to analyze. Then we recorded categorical variables according to their codes on the codebook found on the SAMHSA's website. Furthermore, we created a new variable allied with all drugs and all drugs. The first variable represents the sum of the use of marijuana, cocaine, crack, heroin, hallucinogens, inhalants, and meth in a year, while all drugs represent whether the participant has consumed either of the aforementioned drugs.



```
[16] # Convert to "category" type
     df[['ASDSHOM2','ASDSWRK2','ASDSREL2','ASDSSOC2','ASDSOVL2','IRSEX','IREDUHIGHST2','CATAGE','NEWRACE2','IRWRKSTAT','IRPINC3','INCOME','PDEN10','COUTYP4']] =
```

ALCCAT is alcohol consumption (Yes -> 1 or No -> 0). '0' when IRALCFY = 0 & '1' when IRALCFY is non-zero

ALLDGSCAT is alcohol consumption (Yes -> 1 or No -> 0). '0' when ALLDGS = 0 & '1' when ALLDGS is non-zero

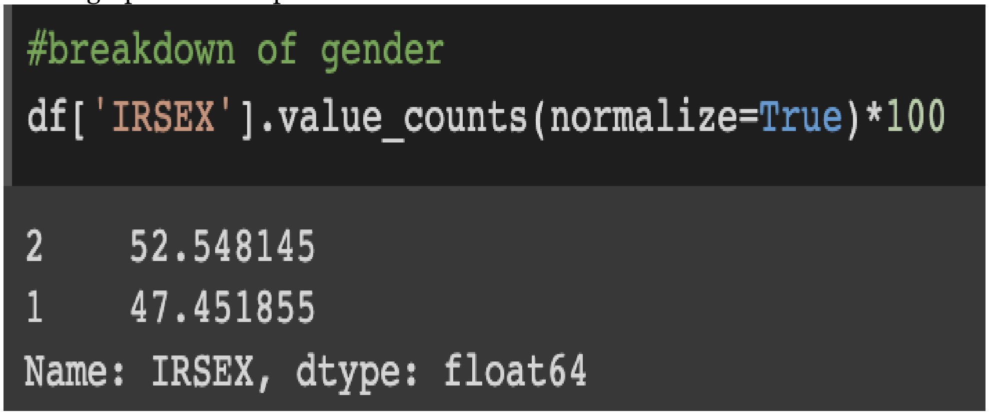
```
# Creating ALCCAT and ALLDGSCAT
def mickey(var):
  if var != 0:
    return 1
  else:
    return 0

df['ALCCAT'] = df['IRALCFY']
df['ALLDGSCAT'] = df['ALLDGS']
df['ALCCAT'] = df['ALCCAT'].apply(mickey)
df['ALLDGSCAT'] = df['ALLDGSCAT'].apply(mickey)
```

4 Data Analysis

4.1 Descriptive Statistics

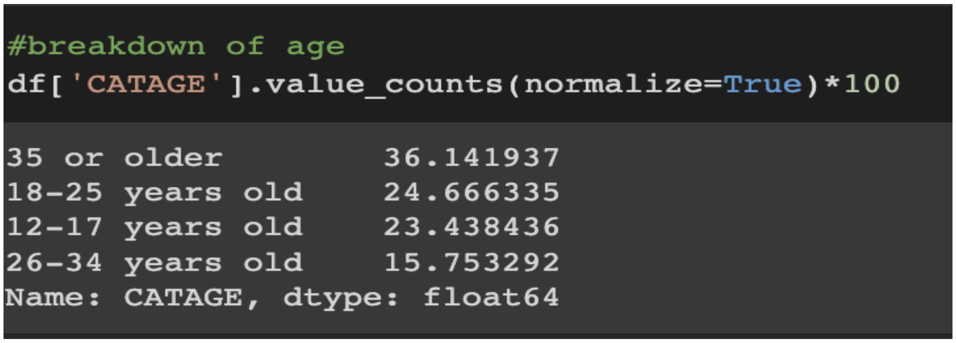
We used the describe() function to describe and summarize data. Refer to figure 1 for a demographical description of our data.

Fig 1. Demographical description of the data



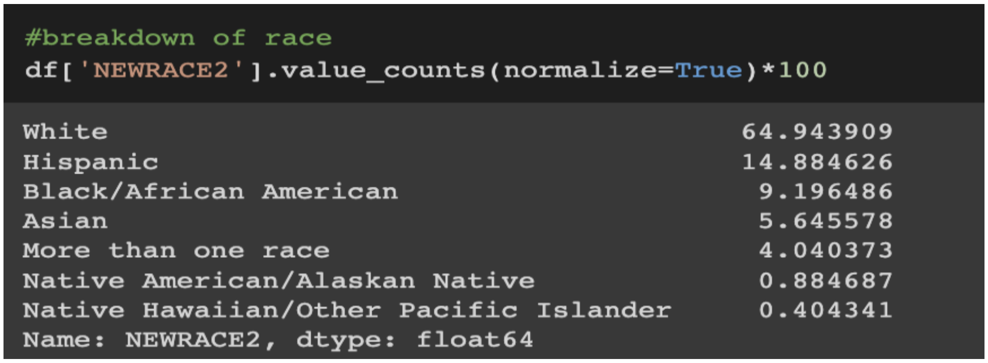
```
#breakdown of gender
df['IRSEX'].value_counts(normalize=True)*100
```

```
2    52.548145
1    47.451855
Name: IRSEX, dtype: float64
```



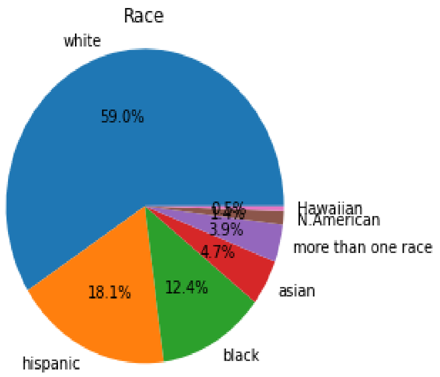
```
#breakdown of age
df['CATAGE'].value_counts(normalize=True)*100
```

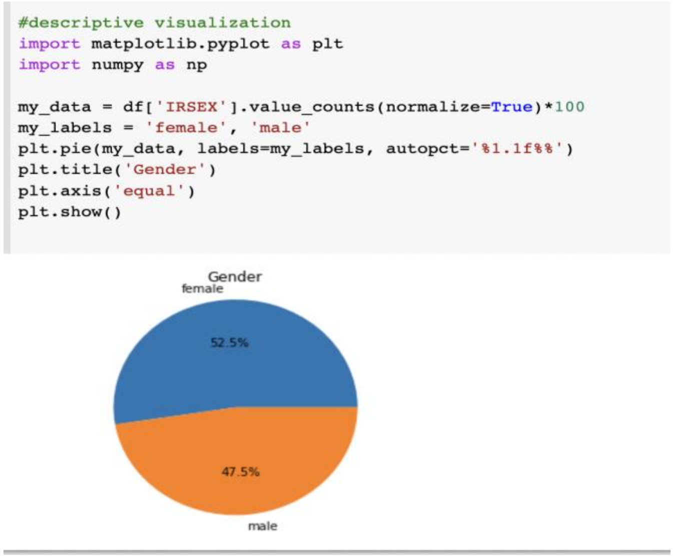
```
35 or older          36.141937
18-25 years old      24.666335
12-17 years old      23.438436
26-34 years old      15.753292
Name: CATAGE, dtype: float64
```



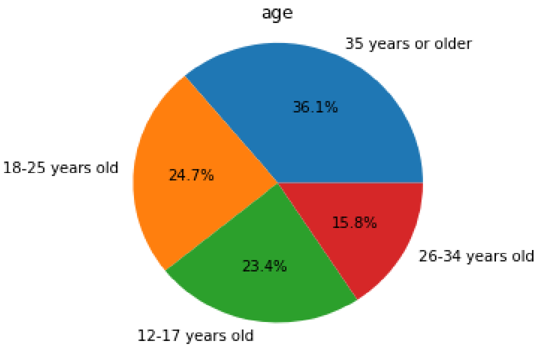
```
#breakdown of race
df['NEWRACE2'].value_counts(normalize=True)*100
```

```
White                                        64.943909
Hispanic                                     14.884626
Black/African American                        9.196486
Asian                                         5.645578
More than one race                            4.040373
Native American/Alaskan Native                0.884687
Native Hawaiian/Other Pacific Islander        0.404341
Name: NEWRACE2, dtype: float64
```



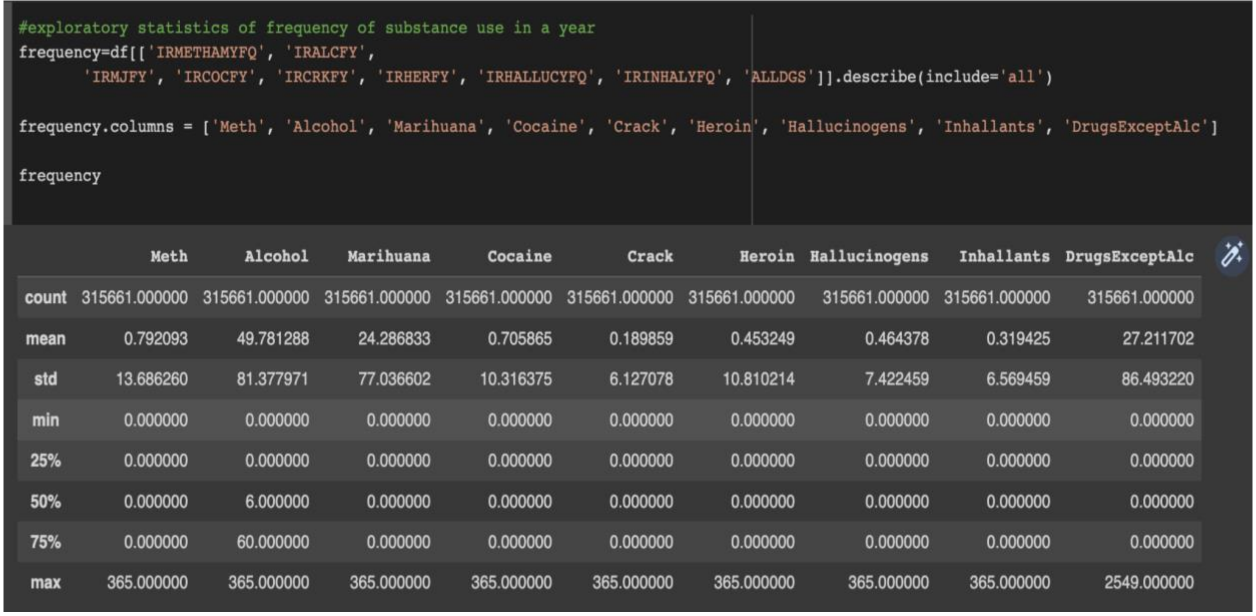


```
#descriptive visualization
import matplotlib.pyplot as plt
import numpy as np

my_data = df['IRSEX'].value_counts(normalize=True)*100
my_labels = 'female', 'male'
plt.pie(my_data, labels=my_labels, autopct='%1.1f%%')
plt.title('Gender')
plt.axis('equal')
plt.show()
```



With regards to the frequency of drug use, refer to figure 2.

Fig 2. Description of data with regards to the frequency of drug use



```
#exploratory statistics of frequency of substance use in a year
frequency=df[['IRMETHAMYFQ', 'IRALCFY',
      'IRMJFY', 'IRCOCFY', 'IRCRKFY', 'IRHERFY', 'IRHALLUCYFQ', 'IRINHALYFQ', 'ALLDGS']].describe(include='all')

frequency.columns = ['Meth', 'Alcohol', 'Marihuana', 'Cocaine', 'Crack', 'Heroin', 'Hallucinogens', 'Inhallants', 'DrugsExceptAlc']

frequency
```

```
                Meth        Alcohol      Marihuana        Cocaine          Crack         Heroin  Hallucinogens     Inhallants  DrugsExceptAlc
count  315661.000000  315661.000000  315661.000000  315661.000000  315661.000000  315661.000000  315661.000000  315661.000000   315661.000000
mean        0.792093      49.781288      24.286833       0.705865       0.189859       0.453249       0.464378       0.319425       27.211702
std        13.686260      81.377971      77.036602      10.316375       6.127078      10.810214       7.422459       6.569459       86.493220
min         0.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.000000        0.000000
25%         0.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.000000        0.000000
50%         0.000000       6.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.000000        0.000000
75%         0.000000      60.000000       0.000000       0.000000       0.000000       0.000000       0.000000       0.000000        0.000000
max       365.000000     365.000000     365.000000     365.000000     365.000000     365.000000     365.000000     365.000000     2549.000000
```

With regards to never/ever using a drug, refer to figure 3

Fig 3

	Alcohol	MJ	Cocaine	Crack	Heroin	Hall.	Inh.	Crack	DrugsExceptAlc
Never	25.48%	54.88%	97.81%	97.53%	98.24%	84.37%	89.74%	95.73%	51.99%
Ever	74.52%	45.12%	2.19%	2.47%	1.76%	15.63%	10.26%	4.27%	48.01%

4.2 Exploratory Data Analysis

df.shape shows that we have 315661 rows and 48 columns.

We have converted all the independent variables like Ethnicity, Age, Gender, Employment, Education, Population density, Health Insurance, Annual household income,

Government Assistance, Perception of unmet need for mental health, Past year major depression, and Level of functioning disability from depression into categorical variables.

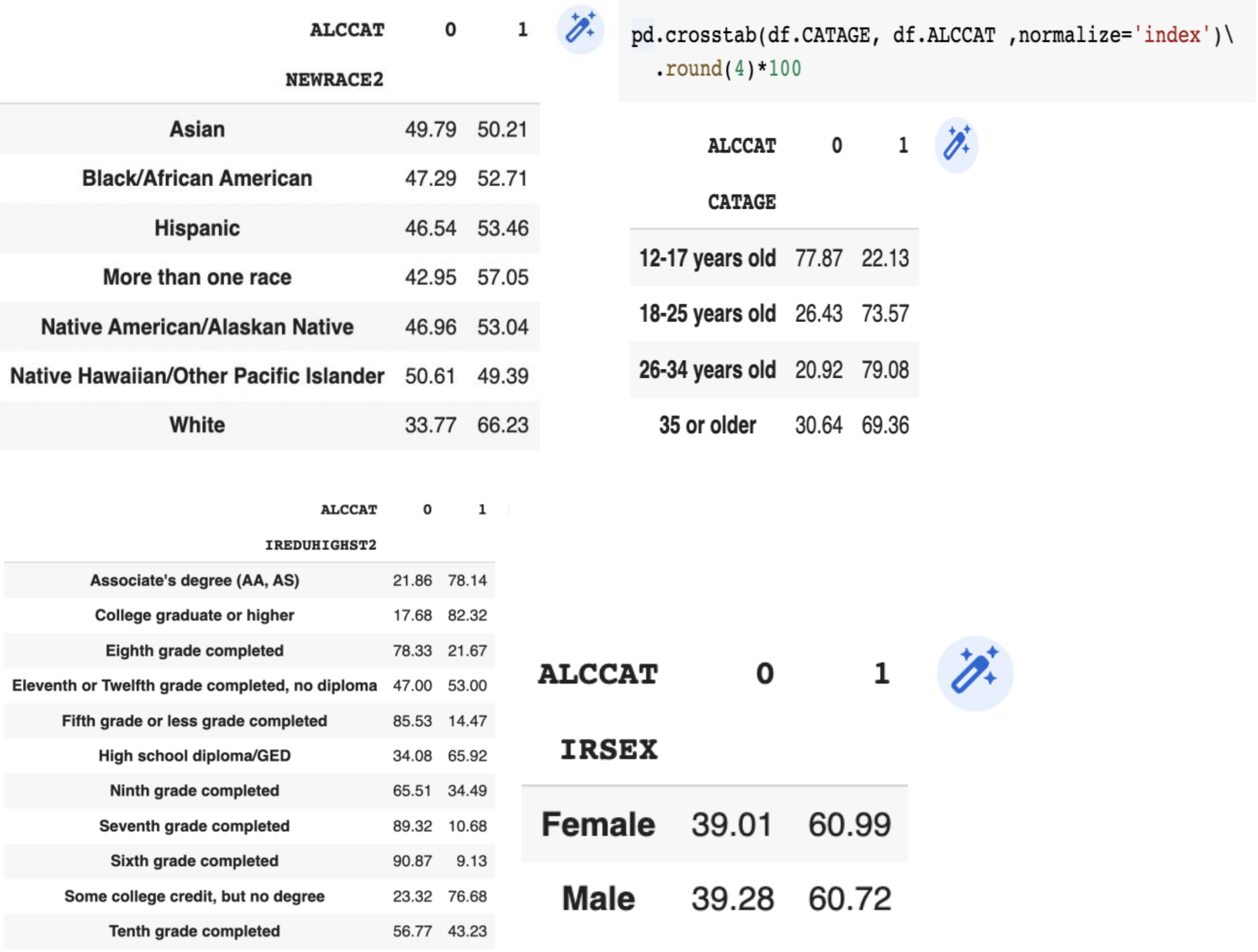
We created 2 dependent frequency variables that are ALCCAT and ALLDGSCAT ALCCAT is alcohol consumption (Yes - 1 or No- 0). ’0’ when IRALCFY is zero and ’1’ when IRALCFY is non-zero

ALLDGSCAT is alcohol consumption (Yes - 1 or No - 0). ’0’ when ALLDGS is zero and ’1’ when ALLDGS is non-zero

By using the describe() function from the Python package Pandas we have obtained multiple summary statistics like the average, standard deviation, median, etc. Due to most of our sample not using substances that are not alcohol and marijuana we created a new variable (ALLDGS) that included all these substances.

Using crosstabs from the pandas package, we created tables to view the percentage of participants who consumed alcohol (fig 3) and drugs (fig 4).

Figure 3. Percentage of participants who consumed alcohol



ALCCAT	0	1
NEWRACE2		
Asian	49.79	50.21
Black/African American	47.29	52.71
Hispanic	46.54	53.46
More than one race	42.95	57.05
Native American/Alaskan Native	46.96	53.04
Native Hawaiian/Other Pacific Islander	50.61	49.39
White	33.77	66.23

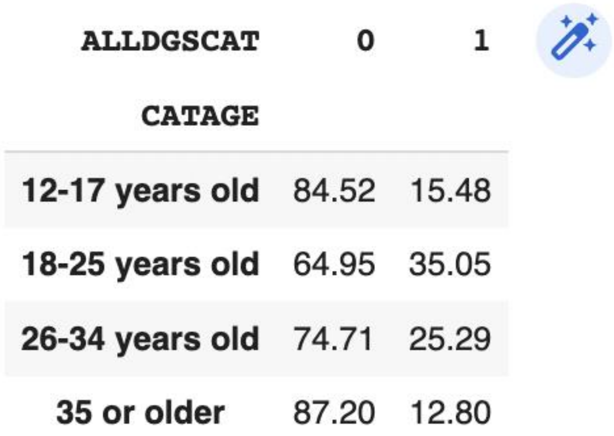
```
pd.crosstab(df.CATAGE, df.ALCCAT ,normalize='index')\
  .round(4)*100
```

ALCCAT	0	1
CATAGE		
12-17 years old	77.87	22.13
18-25 years old	26.43	73.57
26-34 years old	20.92	79.08
35 or older	30.64	69.36

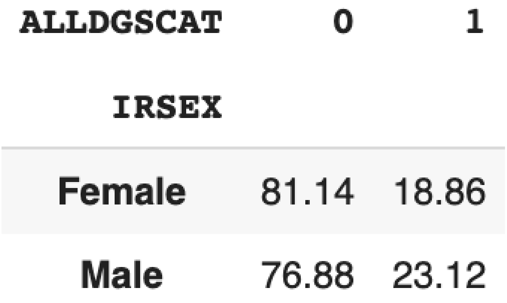
ALCCAT	0	1
IREDUHIGHST2		
Associate's degree (AA, AS)	21.86	78.14
College graduate or higher	17.68	82.32
Eighth grade completed	78.33	21.67
Eleventh or Twelfth grade completed, no diploma	47.00	53.00
Fifth grade or less grade completed	85.53	14.47
High school diploma/GED	34.08	65.92
Ninth grade completed	65.51	34.49
Seventh grade completed	89.32	10.68
Sixth grade completed	90.87	9.13
Some college credit, but no degree	23.32	76.68
Tenth grade completed	56.77	43.23

ALCCAT	0	1
IRSEX		
Female	39.01	60.99
Male	39.28	60.72

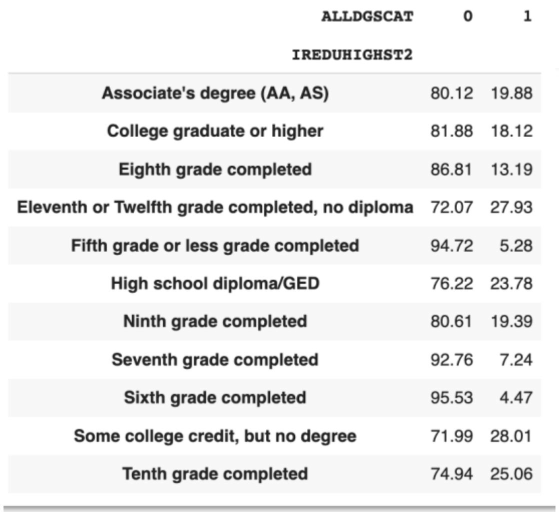
Figure 3.1. Percentage of participants who consumed drugs



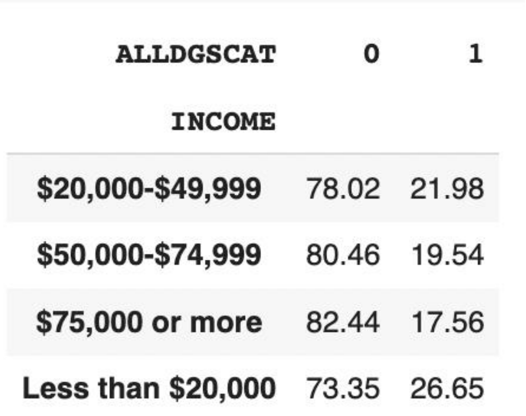
ALLDGSCAT	0	1
CATAGE		
12-17 years old	84.52	15.48
18-25 years old	64.95	35.05
26-34 years old	74.71	25.29
35 or older	87.20	12.80



ALLDGSCAT	0	1
IRSEX		
Female	81.14	18.86
Male	76.88	23.12



ALLDGSCAT	0	1
IREDUHIGHST2		
Associate's degree (AA, AS)	80.12	19.88
College graduate or higher	81.88	18.12
Eighth grade completed	86.81	13.19
Eleventh or Twelfth grade completed, no diploma	72.07	27.93
Fifth grade or less grade completed	94.72	5.28
High school diploma/GED	76.22	23.78
Ninth grade completed	80.61	19.39
Seventh grade completed	92.76	7.24
Sixth grade completed	95.53	4.47
Some college credit, but no degree	71.99	28.01
Tenth grade completed	74.94	25.06

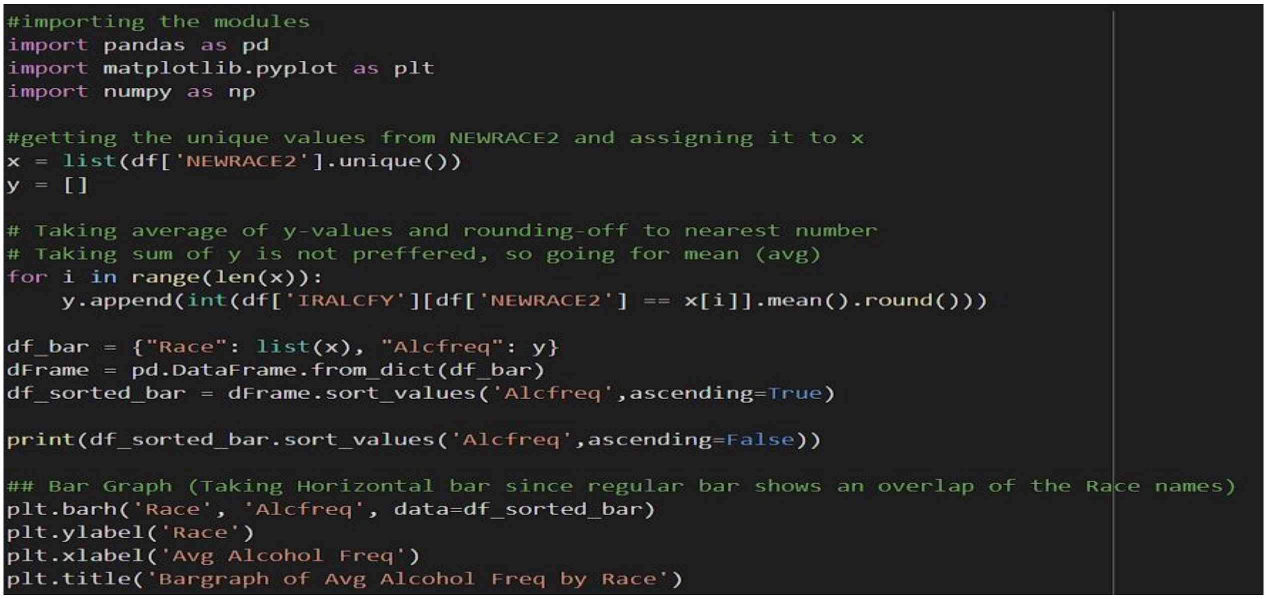


ALLDGSCAT	0	1
INCOME		
$20,000-$49,999	78.02	21.98
$50,000-$74,999	80.46	19.54
$75,000 or more	82.44	17.56
Less than $20,000	73.35	26.65

4.3 Visualization

Using matplotlib we created bar graphs of average drug frequency use and alcohol frequency use per race, age group, education, and income in order to start exploring relationships in our dataset.

Fig. 4 Code for plotting bar graph of average alcohol frequency by Race



```
#importing the modules
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

#getting the unique values from NEWRACE2 and assigning it to x
x = list(df['NEWRACE2'].unique())
y = []

# Taking average of y-values and rounding-off to nearest number
# Taking sum of y is not preffered, so going for mean (avg)
for i in range(len(x)):
    y.append(int(df['IRALCFY'][df['NEWRACE2'] == x[i]].mean().round()))

df_bar = {"Race": list(x), "Alcfreq": y}
dFrame = pd.DataFrame.from_dict(df_bar)
df_sorted_bar = dFrame.sort_values('Alcfreq',ascending=True)

print(df_sorted_bar.sort_values('Alcfreq',ascending=False))

## Bar Graph (Taking Horizontal bar since regular bar shows an overlap of the Race names)
plt.barh('Race', 'Alcfreq', data=df_sorted_bar)
plt.ylabel('Race')
plt.xlabel('Avg Alcohol Freq')
plt.title('Bargraph of Avg Alcohol Freq by Race')
```

Figure 4.1 Bargraph of average alcohol frequency by Race

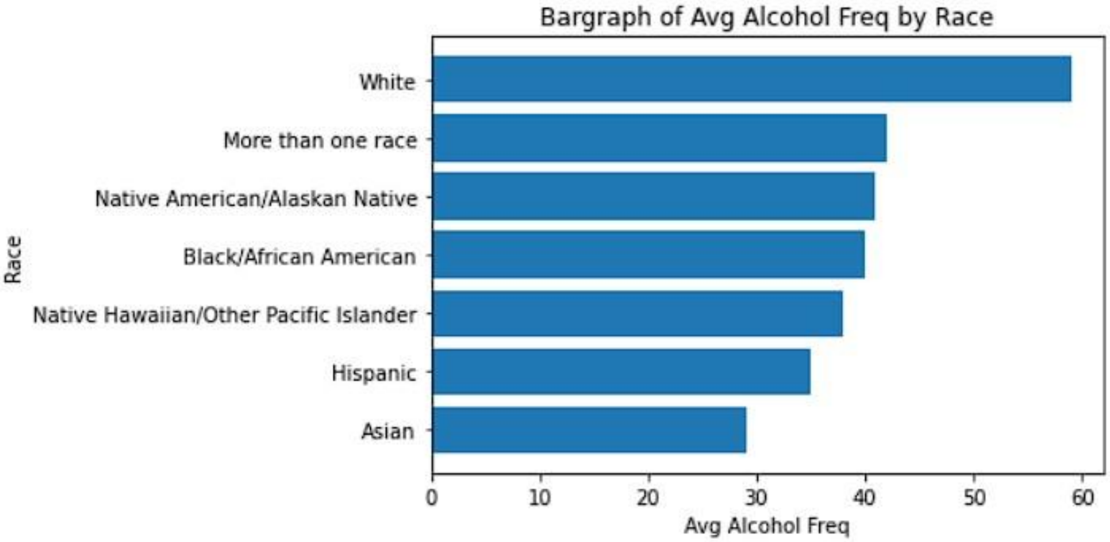


Figure 4.2 Bargraph of average alcohol frequency by Age group

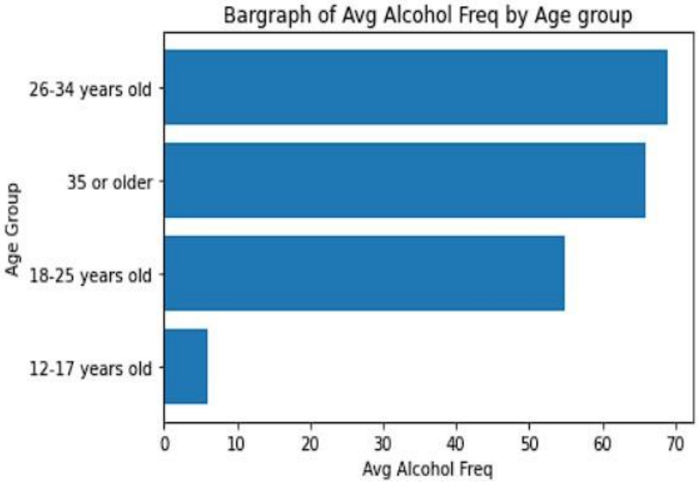


Figure 4.3 Bargraph of average alcohol frequency by Total Income

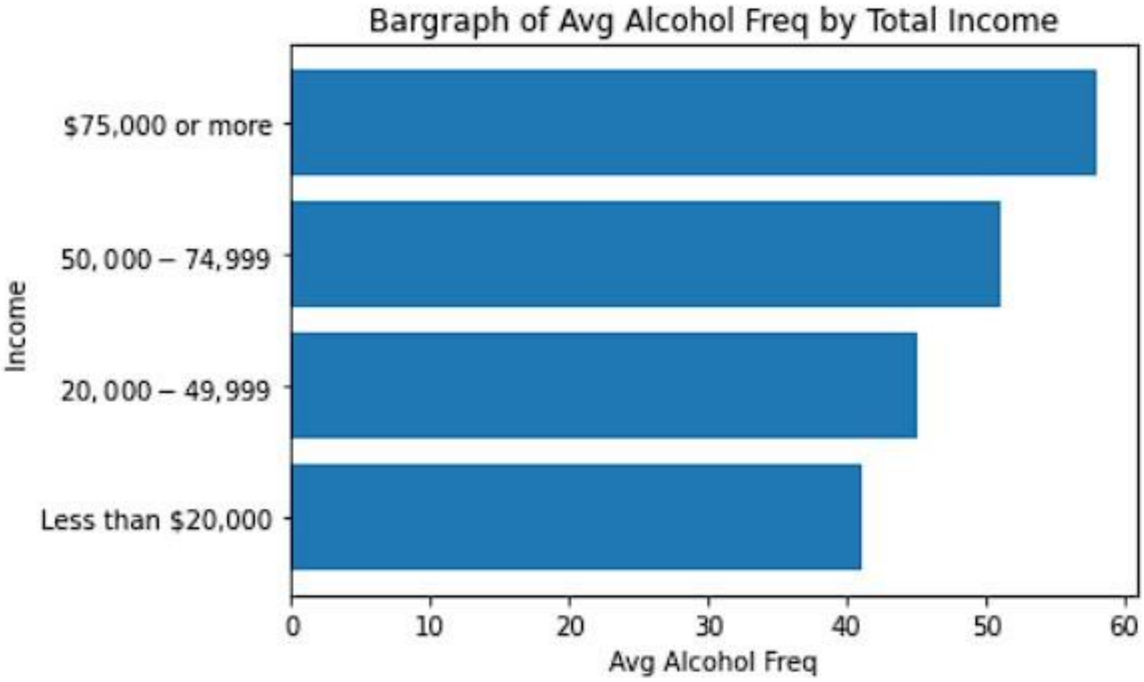


Figure 4.4 Bargraph of average alcohol frequency by Level of Education

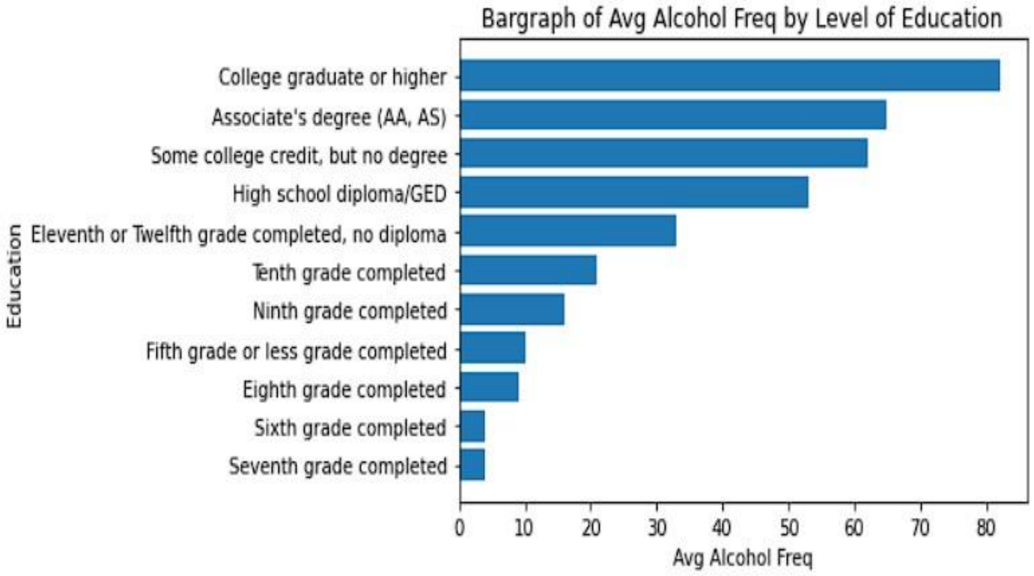


Figure 4.5 Bargraph of average All drugs frequency by Age group

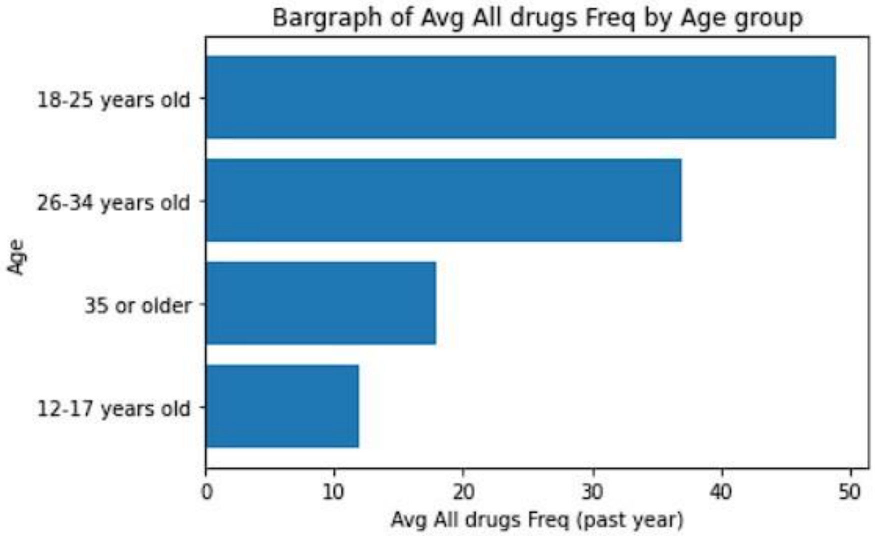


Figure 4.6 Bargraph of average All drugs frequency by Race

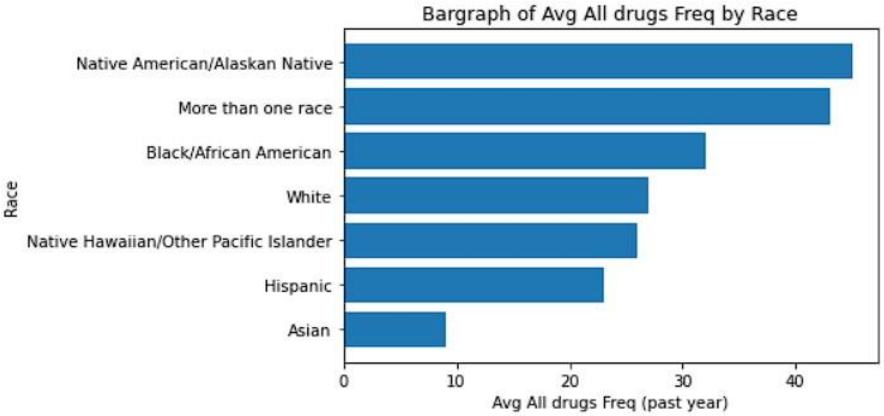


Figure 4.7 Bargraph of average All drugs frequency by Total Income

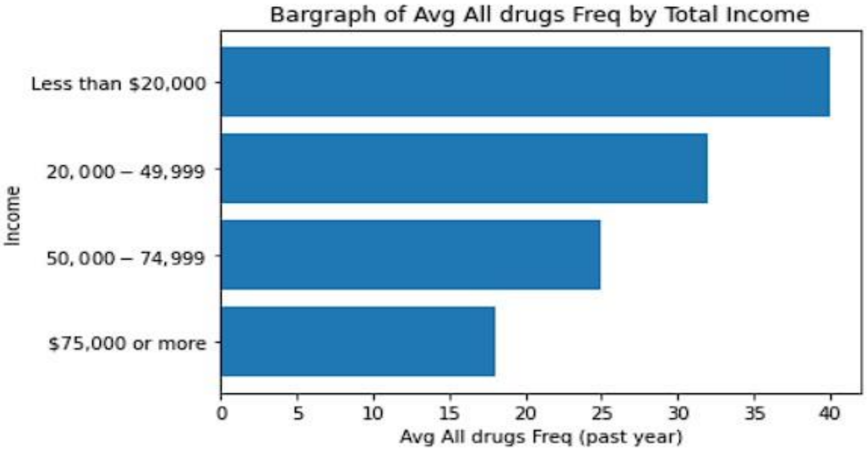
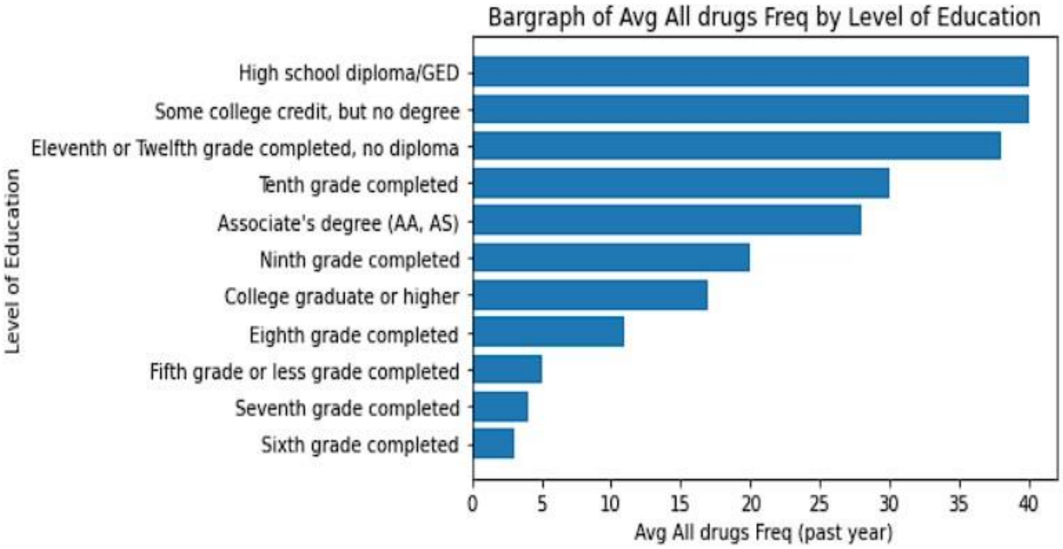
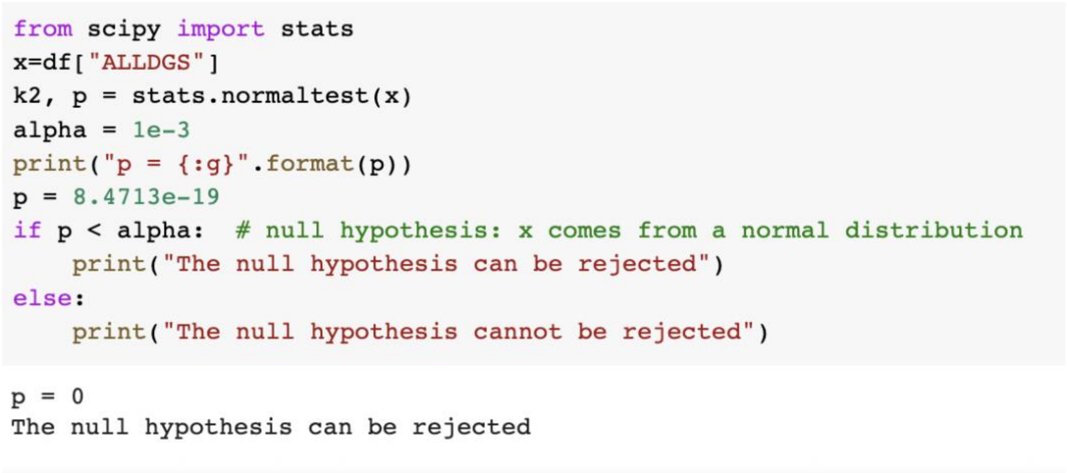


Figure 4.8 Bargraph of average All drugs frequency by Level of Education



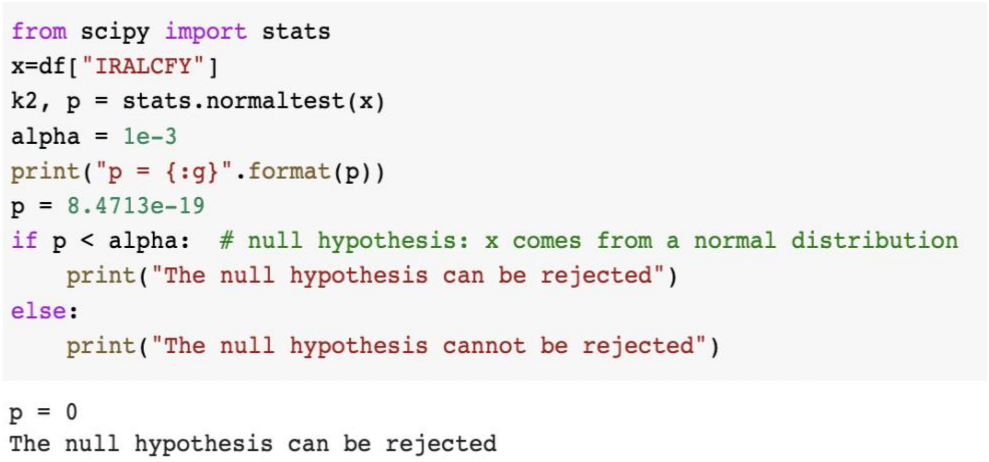
4.4 Normality Test

Using scipy.stats.normaltest we will test if substance use frequency and alcohol use frequency is normally distributed Because neither variable is normally distributed we will use a non-parametric test (Kruskal-Wallis) to see if there is a difference between medians of the groups.



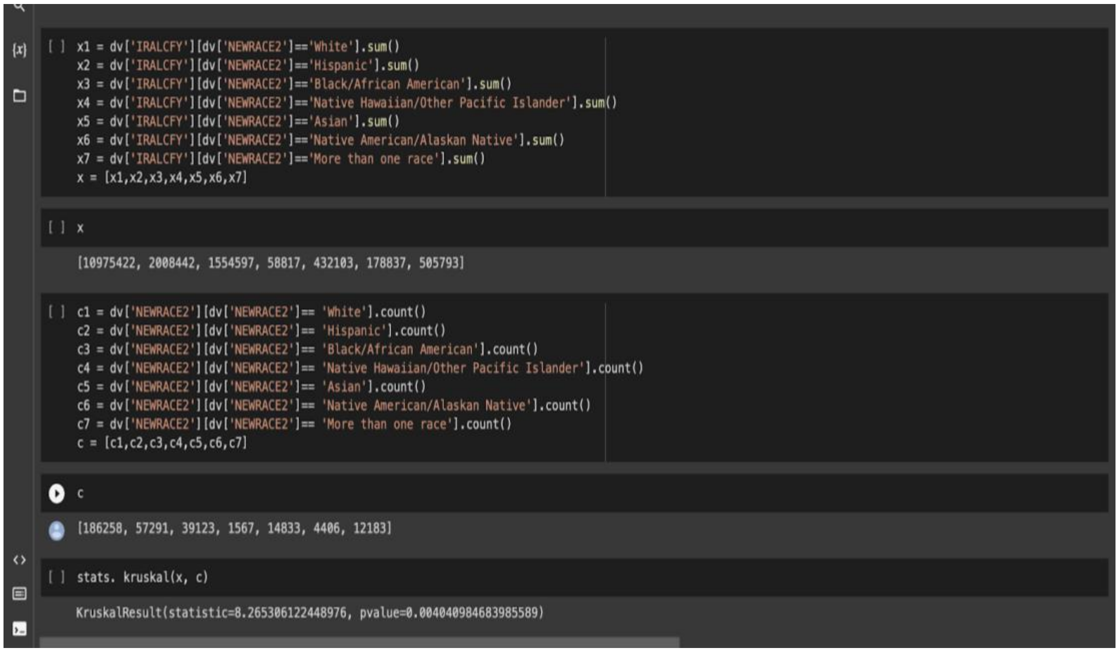
```
from scipy import stats
x=df["ALLDGS"]
k2, p = stats.normaltest(x)
alpha = 1e-3
print("p = {:g}".format(p))
p = 8.4713e-19
if p < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected")
else:
    print("The null hypothesis cannot be rejected")
```

```
p = 0
The null hypothesis can be rejected
```



```
from scipy import stats
x=df["IRALCFY"]
k2, p = stats.normaltest(x)
alpha = 1e-3
print("p = {:g}".format(p))
p = 8.4713e-19
if p < alpha:  # null hypothesis: x comes from a normal distribution
    print("The null hypothesis can be rejected")
else:
    print("The null hypothesis cannot be rejected")
```

```
p = 0
The null hypothesis can be rejected
```



```
x1 = dv['IRALCFY'][dv['NEWRACE2']=='White'].sum()
x2 = dv['IRALCFY'][dv['NEWRACE2']=='Hispanic'].sum()
x3 = dv['IRALCFY'][dv['NEWRACE2']=='Black/African American'].sum()
x4 = dv['IRALCFY'][dv['NEWRACE2']=='Native Hawaiian/Other Pacific Islander'].sum()
x5 = dv['IRALCFY'][dv['NEWRACE2']=='Asian'].sum()
x6 = dv['IRALCFY'][dv['NEWRACE2']=='Native American/Alaskan Native'].sum()
x7 = dv['IRALCFY'][dv['NEWRACE2']=='More than one race'].sum()
x = [x1,x2,x3,x4,x5,x6,x7]
```

```
x
```

```
[10975422, 2008442, 1554597, 58817, 432103, 178837, 505793]
```

```
c1 = dv['NEWRACE2'][dv['NEWRACE2']== 'White'].count()
c2 = dv['NEWRACE2'][dv['NEWRACE2']== 'Hispanic'].count()
c3 = dv['NEWRACE2'][dv['NEWRACE2']== 'Black/African American'].count()
c4 = dv['NEWRACE2'][dv['NEWRACE2']== 'Native Hawaiian/Other Pacific Islander'].count()
c5 = dv['NEWRACE2'][dv['NEWRACE2']== 'Asian'].count()
c6 = dv['NEWRACE2'][dv['NEWRACE2']== 'Native American/Alaskan Native'].count()
c7 = dv['NEWRACE2'][dv['NEWRACE2']== 'More than one race'].count()
c = [c1,c2,c3,c4,c5,c6,c7]
```

```
c
```

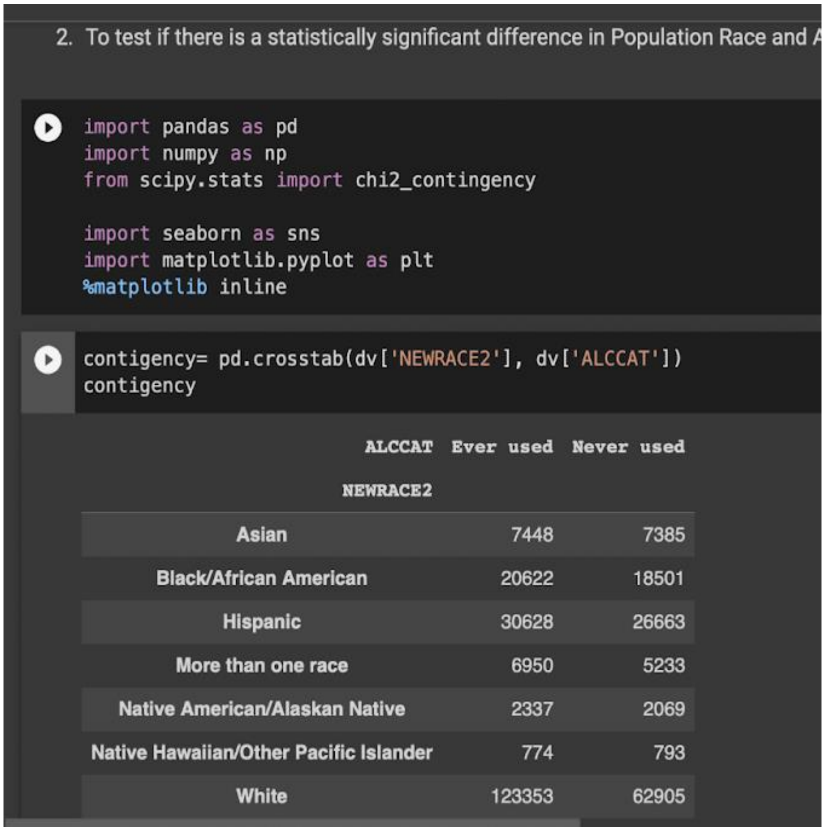
```
[186258, 57291, 39123, 1567, 14833, 4406, 12183]
```

```
stats. kruskal(x, c)
```

```
KruskalResult(statistic=8.265306122448976, pvalue=0.004040984683985589)
```

Based on the p-value (<.05) we found that with regard to alcohol frequency and drug frequency use, age, education, and race had significant differences between groups.

In order to test differences in our groups between our categorical variables, we applied the Chi-Square test, using the stats package. We found that there was a statistically significant (<.05) difference between age, education, and race with regard to alcohol consumption and drug consumption.

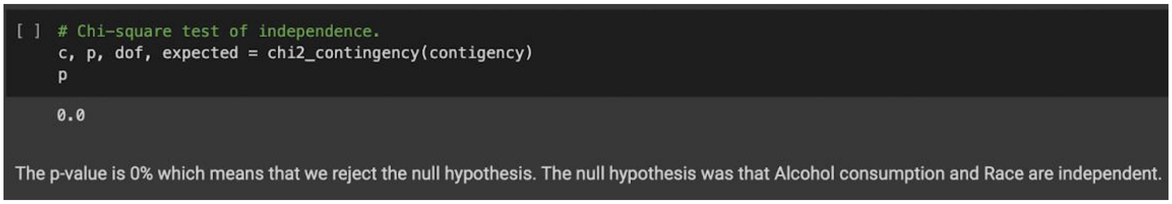


2. To test if there is a statistically significant difference in Population Race and A

```
import pandas as pd
import numpy as np
from scipy.stats import chi2_contingency

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
contigency= pd.crosstab(dv['NEWRACE2'], dv['ALCCAT'])
contigency
```

ALCCAT	Ever used	Never used
NEWRACE2		
Asian	7448	7385
Black/African American	20622	18501
Hispanic	30628	26663
More than one race	6950	5233
Native American/Alaskan Native	2337	2069
Native Hawaiian/Other Pacific Islander	774	793
White	123353	62905



```
[ ] # Chi-square test of independence.
    c, p, dof, expected = chi2_contingency(contigency)
    p

    0.0
```

The p-value is 0% which means that we reject the null hypothesis. The null hypothesis was that Alcohol consumption and Race are independent.

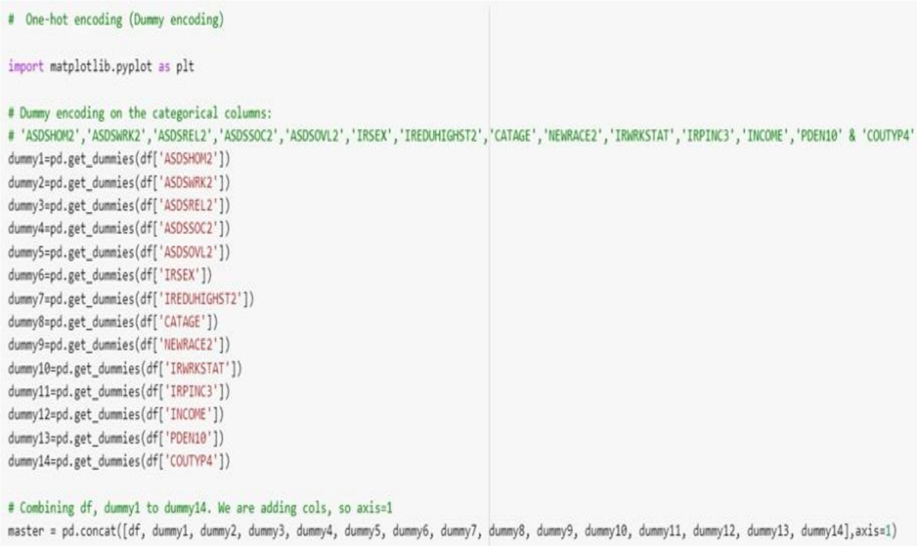
5 Machine Learning Models and Evaluation

By using Machine Learning Models our objectives are the following:

- predict the frequency of yearly alcohol use per participant
- predict the frequency of yearly substance (marihuana, crack, cocaine, meth, heroin, hallucinogens, and inhalants) use per participant
- predict whether participants consumed alcohol
- Predict whether participants consumed any substances (marihuana, crack, cocaine, meth, heroin, hallucinogens, and inhalants)

5.1 Categorical Models

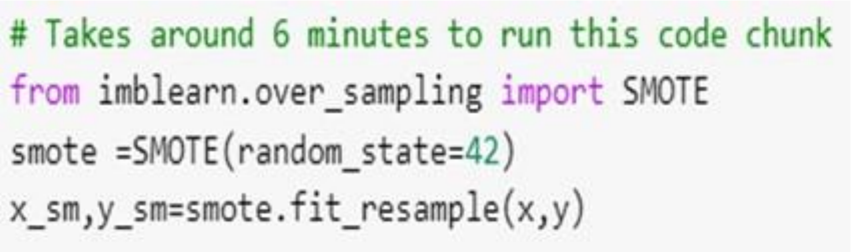
Before implementing the models, we performed one-hot encoding on the categorical independent variables, this converts categories to 0s and 1s, which signifies the presence or absence of the categorical variables, respectively.



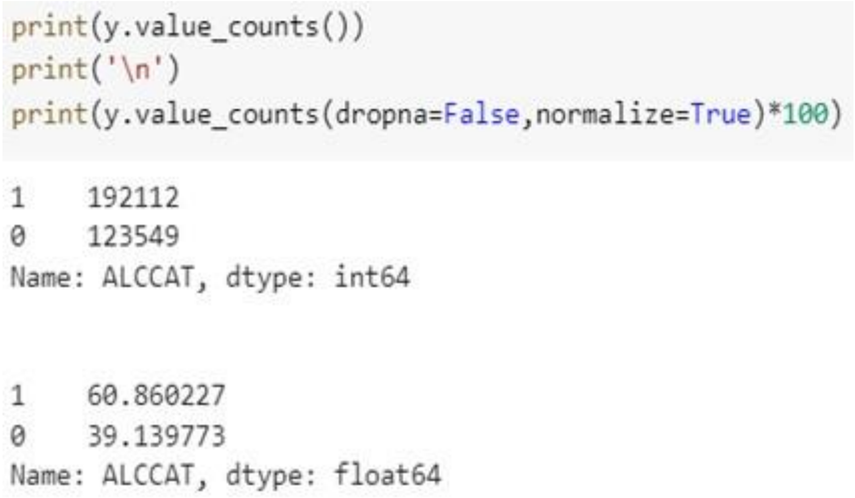
```
#  One-hot encoding (Dummy encoding)

import matplotlib.pyplot as plt

# Dummy encoding on the categorical columns:
# 'ASDSHOM2','ASDSWRK2','ASDSREL2','ASDSSOC2','ASDSOVL2','IRSEX','IREDUHIGHST2','CATAGE','NEWRACE2','IRWRKSTAT','IRPINC3','INCOME','PDEN10' & 'COUTYP4'
dummy1=pd.get_dummies(df['ASDSHOM2'])
dummy2=pd.get_dummies(df['ASDSWRK2'])
dummy3=pd.get_dummies(df['ASDSREL2'])
dummy4=pd.get_dummies(df['ASDSSOC2'])
dummy5=pd.get_dummies(df['ASDSOVL2'])
dummy6=pd.get_dummies(df['IRSEX'])
dummy7=pd.get_dummies(df['IREDUHIGHST2'])
dummy8=pd.get_dummies(df['CATAGE'])
dummy9=pd.get_dummies(df['NEWRACE2'])
dummy10=pd.get_dummies(df['IRWRKSTAT'])
dummy11=pd.get_dummies(df['IRPINC3'])
dummy12=pd.get_dummies(df['INCOME'])
dummy13=pd.get_dummies(df['PDEN10'])
dummy14=pd.get_dummies(df['COUTYP4'])

# Combining df, dummy1 to dummy14. We are adding cols, so axis=1
master = pd.concat([df, dummy1, dummy2, dummy3, dummy4, dummy5, dummy6, dummy7, dummy8, dummy9, dummy10, dummy11, dummy12, dummy13, dummy14],axis=1)
```

Furthermore, by using the SMOTE technique (Chawla et al., 2002)[vii] in order to oversample and improve the specificity and sensitivity of our models. Using the SMOTE-transformed variables, we then create our test and training data.

Using the SMOTE-transformed variables, we then create our test and training data.

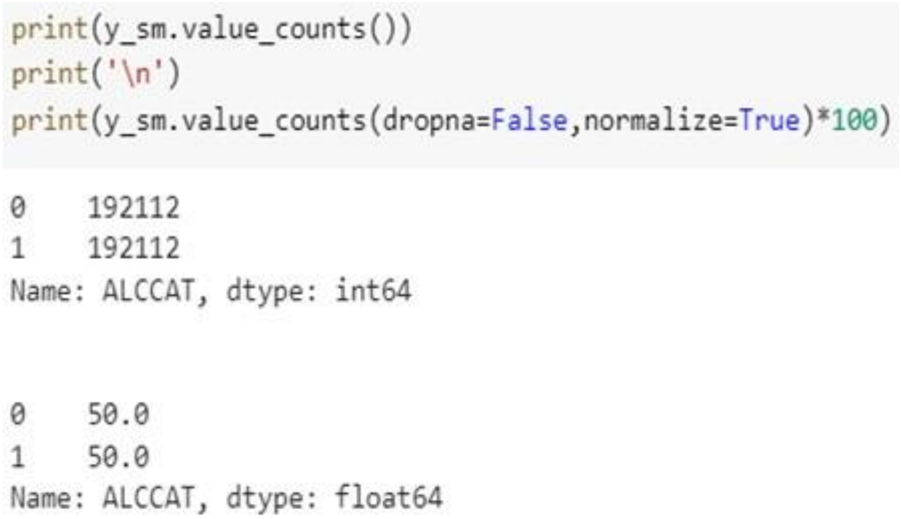


```
# Takes around 6 minutes to run this code chunk
from imblearn.over_sampling import SMOTE
smote =SMOTE(random_state=42)
x_sm,y_sm=smote.fit_resample(x,y)
```



```
print(y.value_counts())
print('\n')
print(y.value_counts(dropna=False,normalize=True)*100)
```

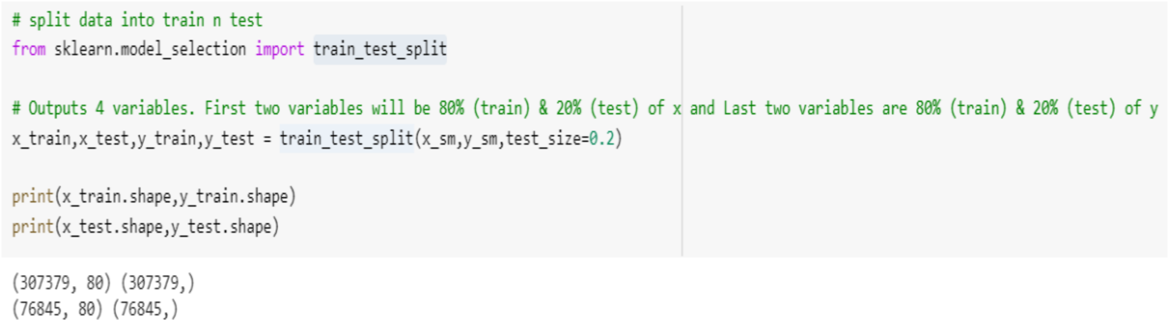
```
1    192112
0    123549
Name: ALCCAT, dtype: int64


1    60.860227
0    39.139773
Name: ALCCAT, dtype: float64
```



```
print(y_sm.value_counts())
print('\n')
print(y_sm.value_counts(dropna=False,normalize=True)*100)
```

```
0    192112
1    192112
Name: ALCCAT, dtype: int64


0    50.0
1    50.0
Name: ALCCAT, dtype: float64
```

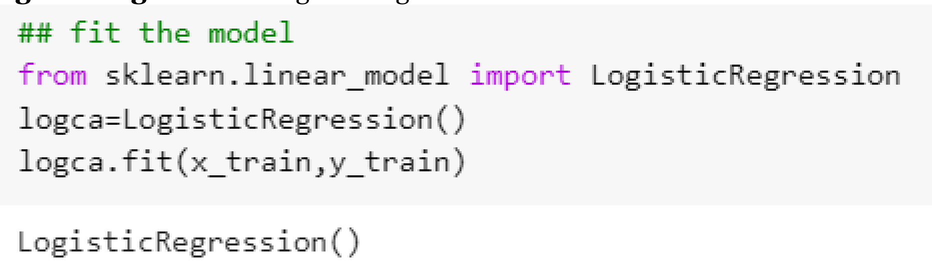
Test-Train Split



```
# split data into train n test
from sklearn.model_selection import train_test_split

# Outputs 4 variables. First two variables will be 80% (train) & 20% (test) of x and Last two variables are 80% (train) & 20% (test) of y
x_train,x_test,y_train,y_test = train_test_split(x_sm,y_sm,test_size=0.2)

print(x_train.shape,y_train.shape)
print(x_test.shape,y_test.shape)
```

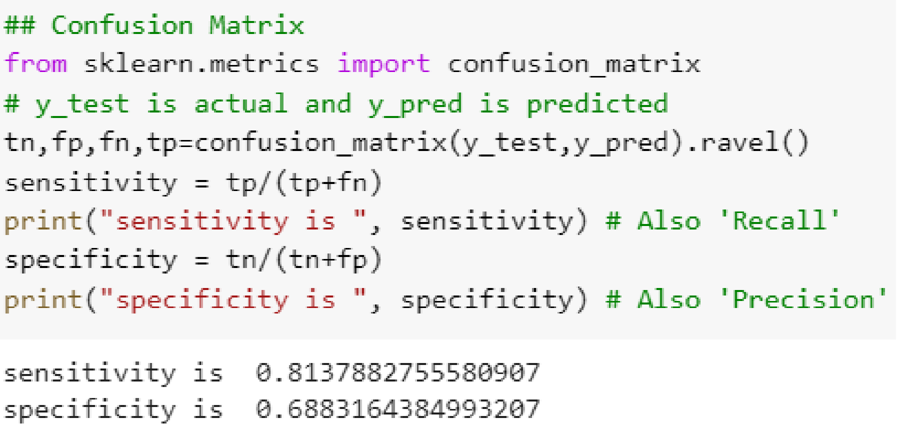
```
(307379, 80) (307379,)
(76845, 80) (76845,)
```

Logistic Regression- Logistic Regression for Alcohol use.



```
## fit the model
from sklearn.linear_model import LogisticRegression
logca=LogisticRegression()
logca.fit(x_train,y_train)
```

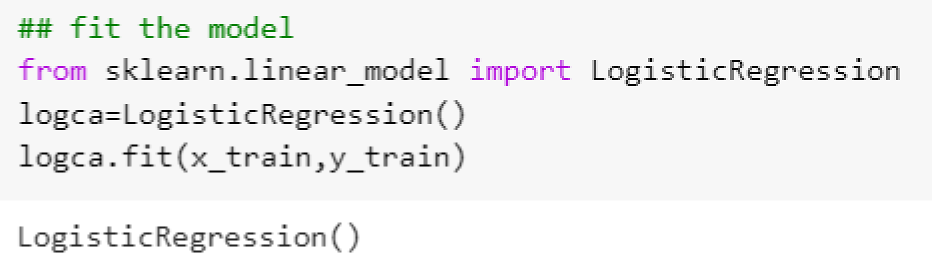
```
LogisticRegression()
```



```
## Confusion Matrix
from sklearn.metrics import confusion_matrix
# y_test is actual and y_pred is predicted
tn,fp,fn,tp=confusion_matrix(y_test,y_pred).ravel()
sensitivity = tp/(tp+fn)
print("sensitivity is ", sensitivity) # Also 'Recall'
specificity = tn/(tn+fp)
print("specificity is ", specificity) # Also 'Precision'
```

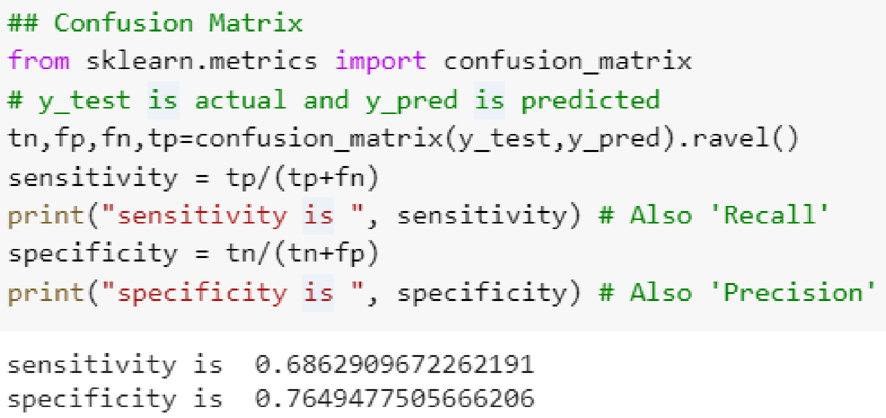
```
sensitivity is  0.8137882755580907
specificity is  0.6883164384993207
```

Logistic Regression for Alcohol use.



```
## fit the model
from sklearn.linear_model import LogisticRegression
logca=LogisticRegression()
logca.fit(x_train,y_train)
```

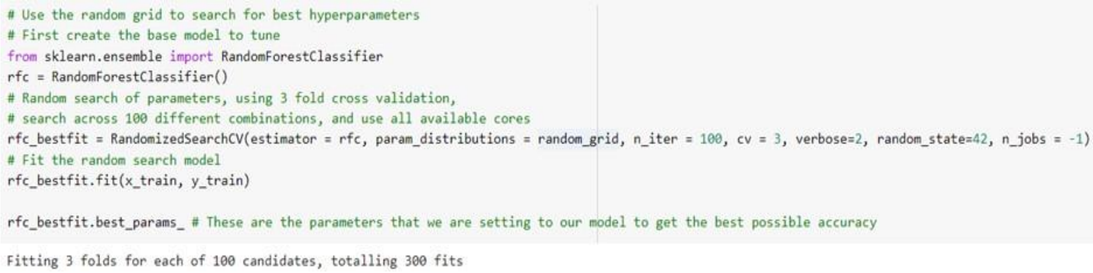
```
LogisticRegression()
```



```
## Confusion Matrix
from sklearn.metrics import confusion_matrix
# y_test is actual and y_pred is predicted
tn,fp,fn,tp=confusion_matrix(y_test,y_pred).ravel()
sensitivity = tp/(tp+fn)
print("sensitivity is ", sensitivity) # Also 'Recall'
specificity = tn/(tn+fp)
print("specificity is ", specificity) # Also 'Precision'
```

```
sensitivity is  0.6862909672262191
specificity is  0.7649477505666206
```

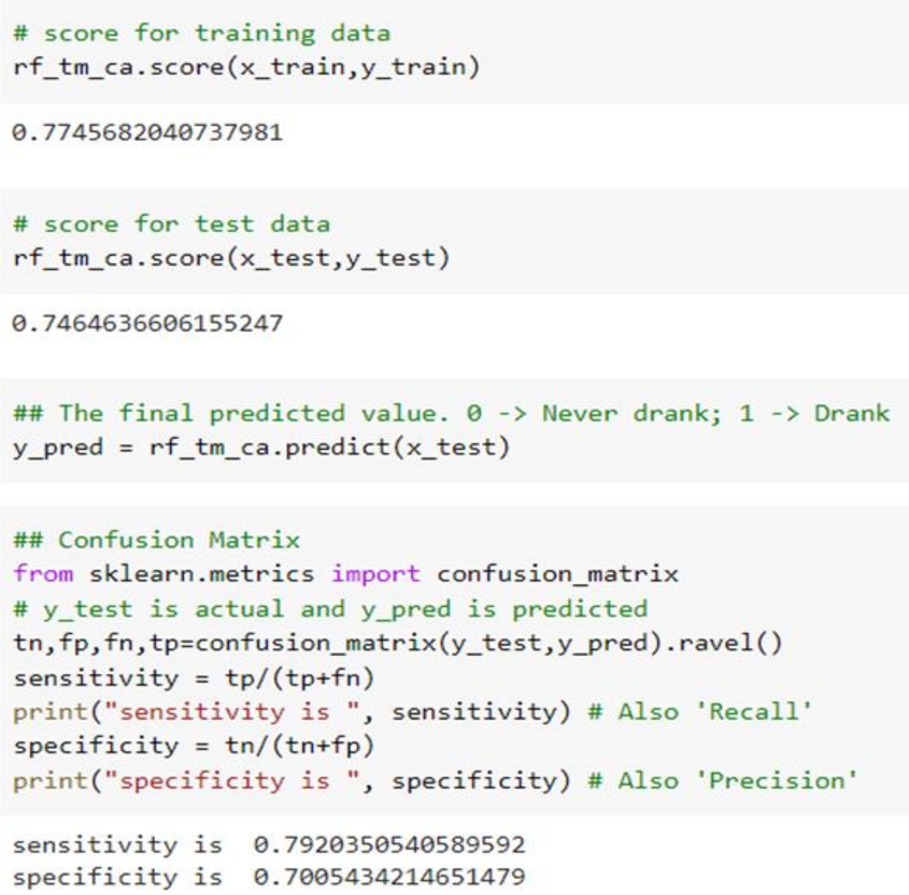
Random forest classifier- Before implementing any models we use hyperparameter tuning in order to find the best parameters for both the Random Forest Classifier and Random forest regression. Fig. 13 Code to find optimal parameters for Alcohol use model



```
# Use the random grid to search for best hyperparameters
# First create the base model to tune
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier()
# Random search of parameters, using 3 fold cross validation,
# search across 100 different combinations, and use all available cores
rfc_bestfit = RandomizedSearchCV(estimator = rfc, param_distributions = random_grid, n_iter = 100, cv = 3, verbose=2, random_state=42, n_jobs = -1)
# Fit the random search model
rfc_bestfit.fit(x_train, y_train)

rfc_bestfit.best_params_ # These are the parameters that we are setting to our model to get the best possible accuracy
```

```
Fitting 3 folds for each of 100 candidates, totalling 300 fits
```

Fig. 14 Random Forest Classifier code and output for Alcohol use



```
# score for training data
rf_tm_ca.score(x_train,y_train)
```

```
0.7745682040737981
```

```
# score for test data
rf_tm_ca.score(x_test,y_test)
```

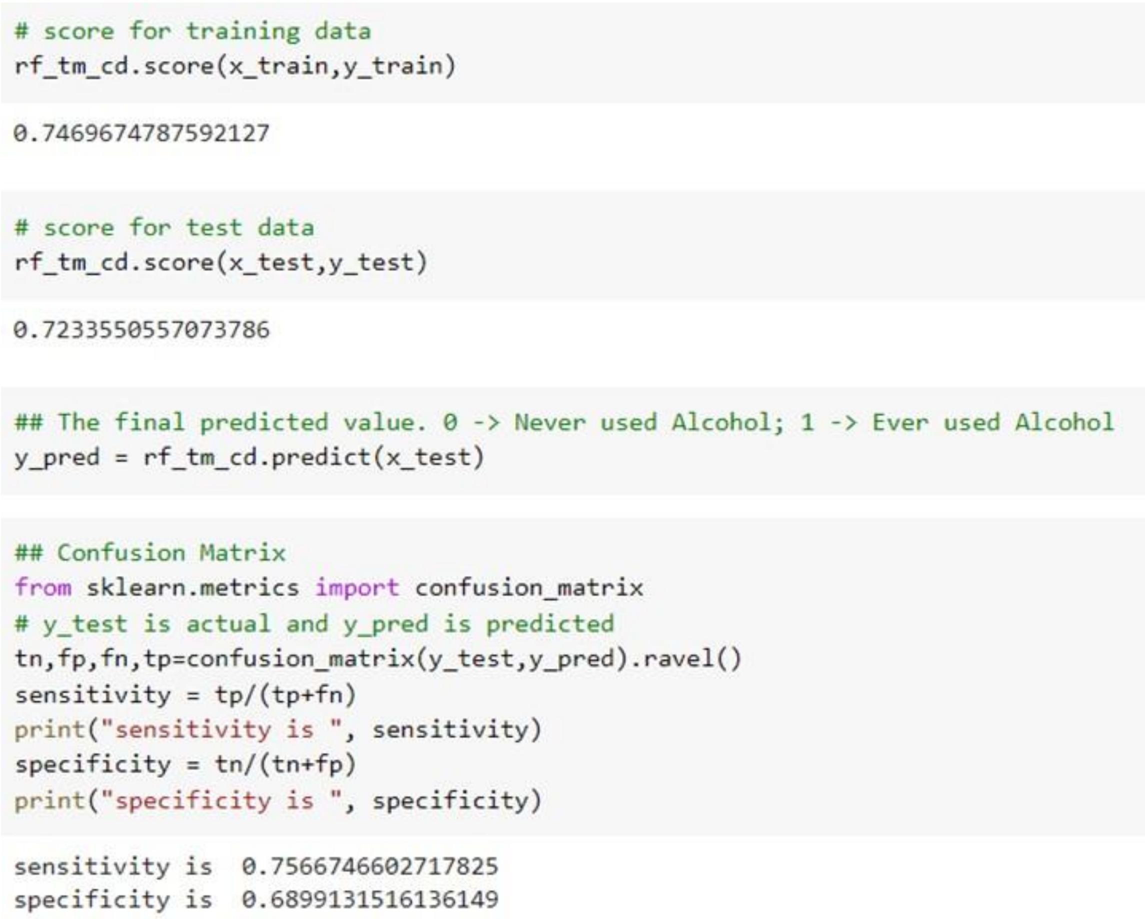
```
0.7464636606155247
```

```
## The final predicted value. 0 -> Never drank; 1 -> Drank
y_pred = rf_tm_ca.predict(x_test)
```

```
## Confusion Matrix
from sklearn.metrics import confusion_matrix
# y_test is actual and y_pred is predicted
tn,fp,fn,tp=confusion_matrix(y_test,y_pred).ravel()
sensitivity = tp/(tp+fn)
print("sensitivity is ", sensitivity) # Also 'Recall'
specificity = tn/(tn+fp)
print("specificity is ", specificity) # Also 'Precision'
```

```
sensitivity is  0.7920350540589592
specificity is  0.7005434214651479
```

Prior to implementing Random Forest Classifier for substance use, we also employ hyperparameter tuning. Fig. 15 Random Forest Classifier code and output for drug use.



```
# score for training data
rf_tm_cd.score(x_train,y_train)
```

```
0.7469674787592127
```

```
# score for test data
rf_tm_cd.score(x_test,y_test)
```

```
0.7233550557073786
```

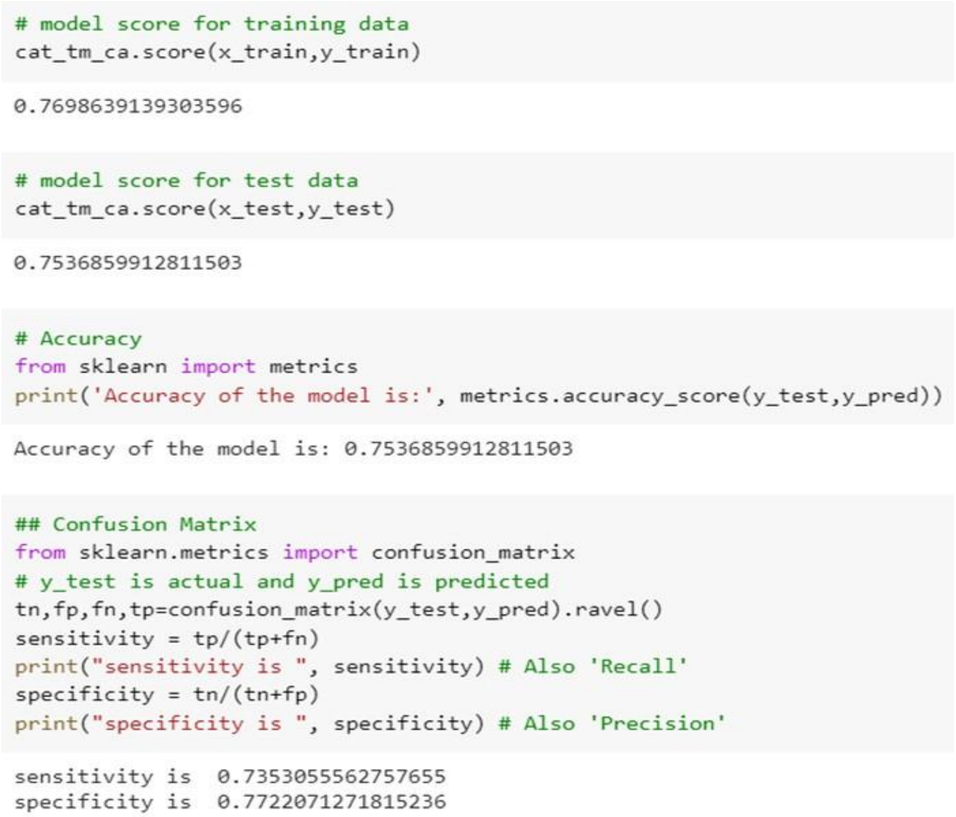
```
## The final predicted value. 0 -> Never used Alcohol; 1 -> Ever used Alcohol
y_pred = rf_tm_cd.predict(x_test)
```

```
## Confusion Matrix
from sklearn.metrics import confusion_matrix
# y_test is actual and y_pred is predicted
tn,fp,fn,tp=confusion_matrix(y_test,y_pred).ravel()
sensitivity = tp/(tp+fn)
print("sensitivity is ", sensitivity)
specificity = tn/(tn+fp)
print("specificity is ", specificity)
```

```
sensitivity is  0.7566746602717825
specificity is  0.6899131516136149
```

CatBoost classifier Prior to using CatBoost classifier, we also implement hyperparameter tuning.

Fig. 16 Code and output for Catboost classifier for alcohol use.



```
# model score for training data
cat_tm_ca.score(x_train,y_train)
```

```
0.7698639139303596
```

```
# model score for test data
cat_tm_ca.score(x_test,y_test)
```

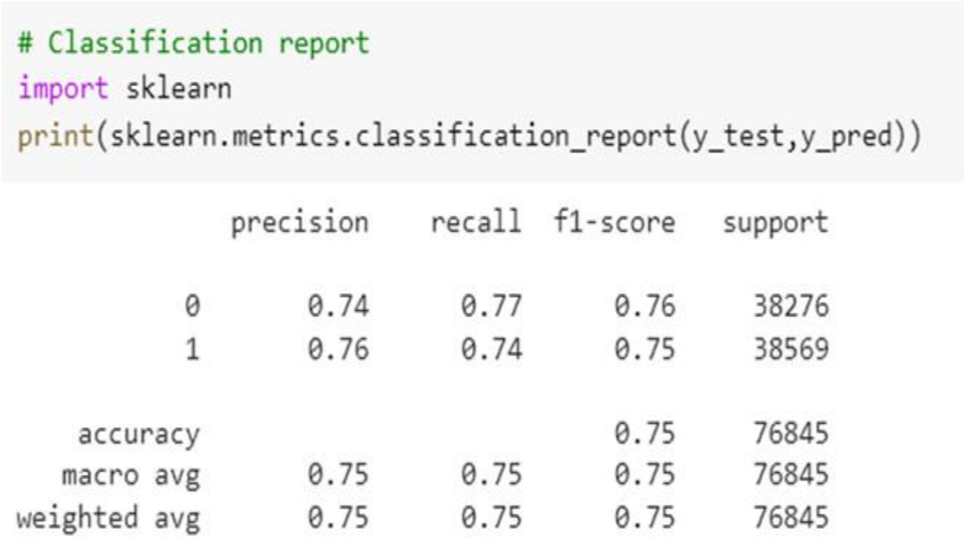
```
0.7536859912811503
```

```
# Accuracy
from sklearn import metrics
print('Accuracy of the model is:', metrics.accuracy_score(y_test,y_pred))
```

```
Accuracy of the model is: 0.7536859912811503
```

```
## Confusion Matrix
from sklearn.metrics import confusion_matrix
# y_test is actual and y_pred is predicted
tn,fp,fn,tp=confusion_matrix(y_test,y_pred).ravel()
sensitivity = tp/(tp+fn)
print("sensitivity is ", sensitivity) # Also 'Recall'
specificity = tn/(tn+fp)
print("specificity is ", specificity) # Also 'Precision'
```

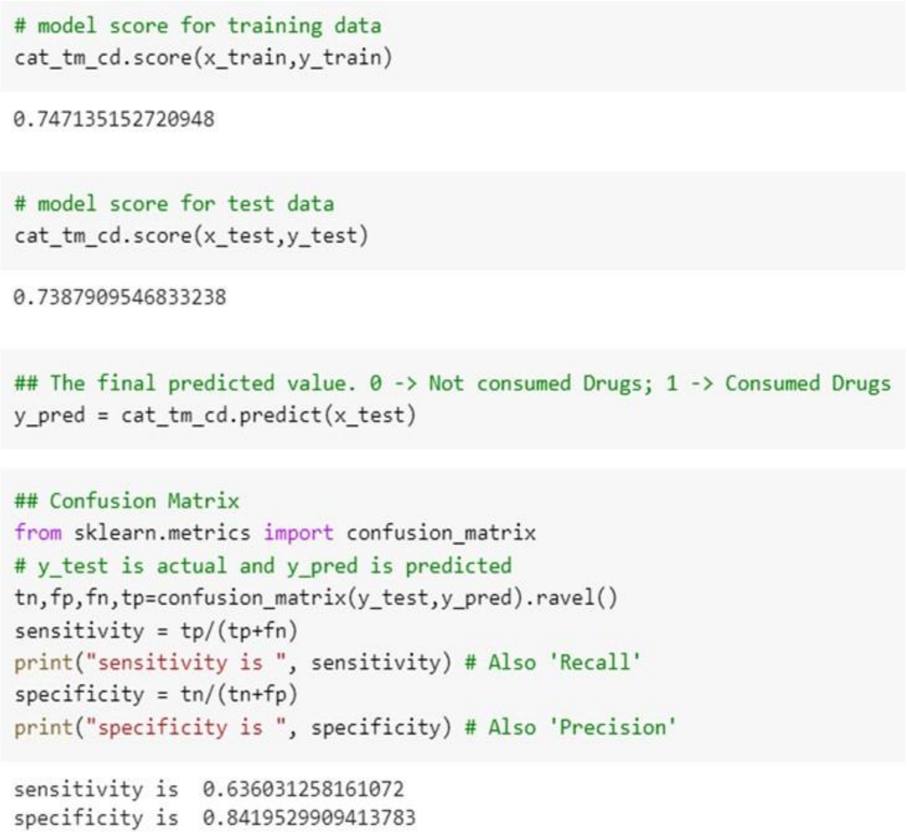
```
sensitivity is  0.7353055562757655
specificity is  0.7722071271815236
```



```
# Classification report
import sklearn
print(sklearn.metrics.classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.74      0.77      0.76     38276
           1       0.76      0.74      0.75     38569

    accuracy                           0.75     76845
   macro avg       0.75      0.75      0.75     76845
weighted avg       0.75      0.75      0.75     76845
```

Fig. 17 Code and output for Catboost classifier for drug use



```
# model score for training data
cat_tm_cd.score(x_train,y_train)
```

```
0.747135152720948
```

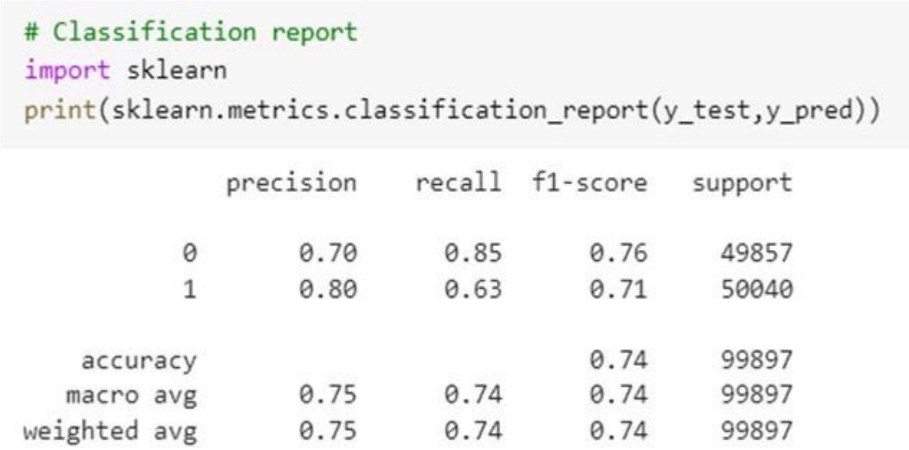
```
# model score for test data
cat_tm_cd.score(x_test,y_test)
```

```
0.7387909546833238
```

```
## The final predicted value. 0 -> Not consumed Drugs; 1 -> Consumed Drugs
y_pred = cat_tm_cd.predict(x_test)
```

```
## Confusion Matrix
from sklearn.metrics import confusion_matrix
# y_test is actual and y_pred is predicted
tn,fp,fn,tp=confusion_matrix(y_test,y_pred).ravel()
sensitivity = tp/(tp+fn)
print("sensitivity is ", sensitivity) # Also 'Recall'
specificity = tn/(tn+fp)
print("specificity is ", specificity) # Also 'Precision'
```

```
sensitivity is  0.636031258161072
specificity is  0.8419529909413783
```

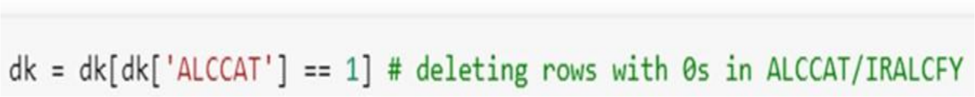


```
# Classification report
import sklearn
print(sklearn.metrics.classification_report(y_test,y_pred))
```

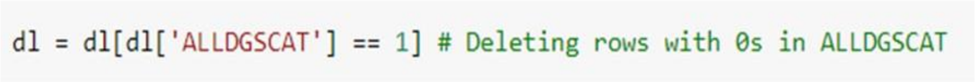
```
              precision    recall  f1-score   support

           0       0.70      0.85      0.76     49857
           1       0.80      0.63      0.71     50040

    accuracy                           0.74     99897
   macro avg       0.75      0.74      0.74     99897
weighted avg       0.75      0.74      0.74     99897
```

5.2 Regression models

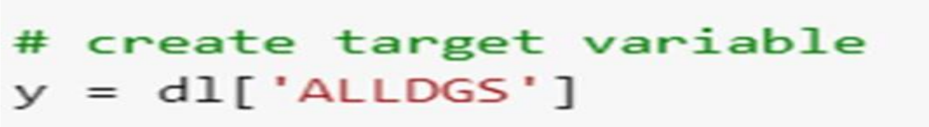
Prior to implementing our models we remove any rows with 0s in the alcohol and drug categorical variables before splitting the dataset intro testing and training (Fig 18). This means that participants who did not consume alcohol or drugs in the past year are excluded from our model. This reduces the skewness of the distribution.



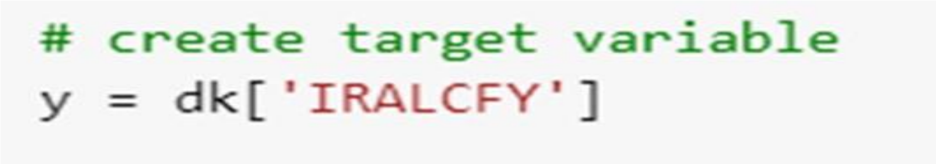
```
dk = dk[dk['ALCCAT'] == 1] # deleting rows with 0s in ALCCAT/IRALCFY
```



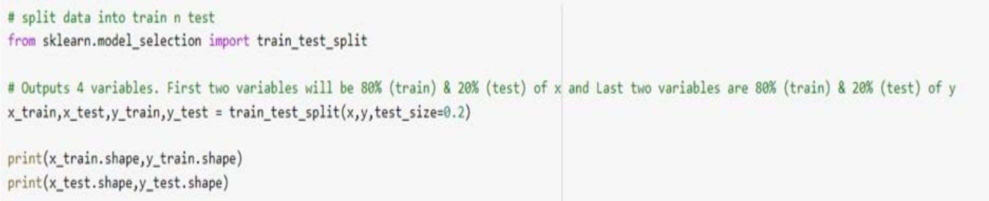
```
dl = dl[dl['ALLDGSCAT'] == 1] # Deleting rows with 0s in ALLDGSCAT
```



```
# create target variable
y = dl['ALLDGS']
```

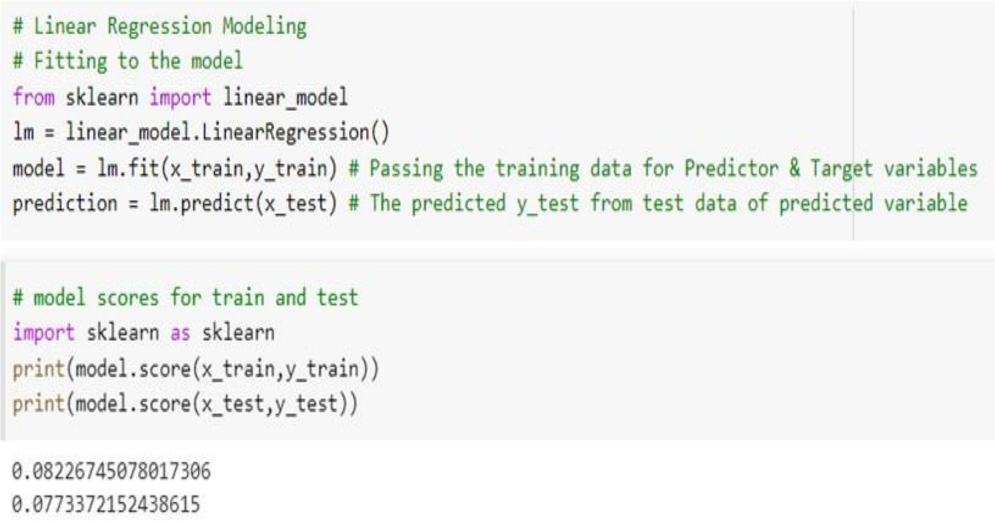


```
# create target variable
y = dk['IRALCFY']
```



```
# split data into train n test
from sklearn.model_selection import train_test_split

# Outputs 4 variables. First two variables will be 80% (train) & 20% (test) of x and Last two variables are 80% (train) & 20% (test) of y
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)

print(x_train.shape,y_train.shape)
print(x_test.shape,y_test.shape)
```

Linear Regression Fig 19. Code for Linear Regression Model of Alcohol Frequency in the past year

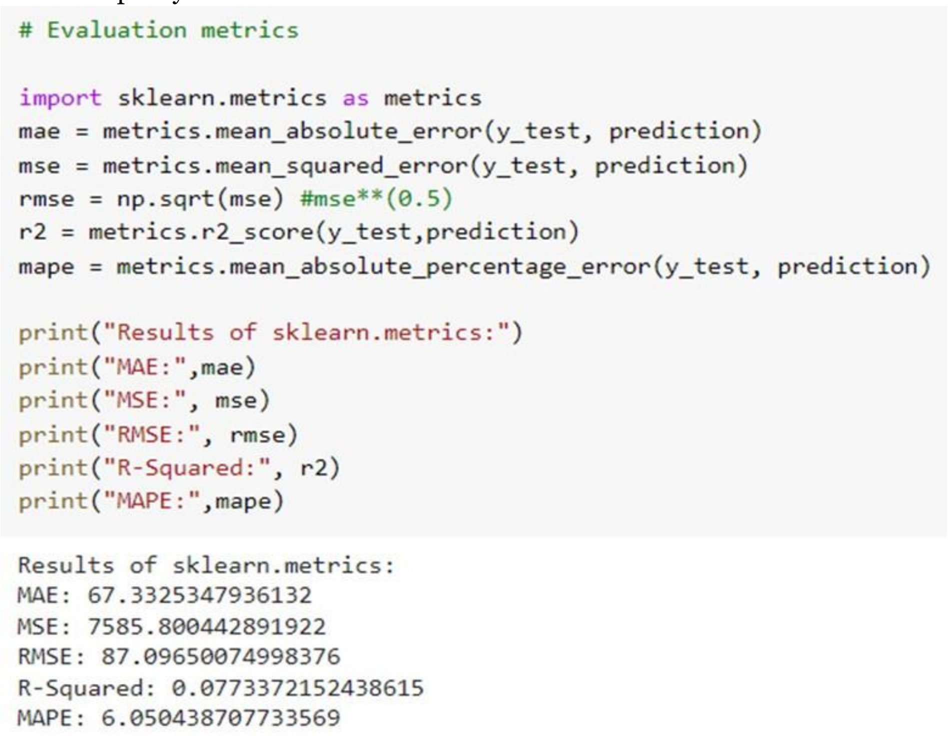


```
# Linear Regression Modeling
# Fitting to the model
from sklearn import linear_model
lm = linear_model.LinearRegression()
model = lm.fit(x_train,y_train) # Passing the training data for Predictor & Target variables
prediction = lm.predict(x_test) # The predicted y_test from test data of predicted variable
```

```
# model scores for train and test
import sklearn as sklearn
print(model.score(x_train,y_train))
print(model.score(x_test,y_test))
```

```
0.08226745078017306
0.0773372152438615
```

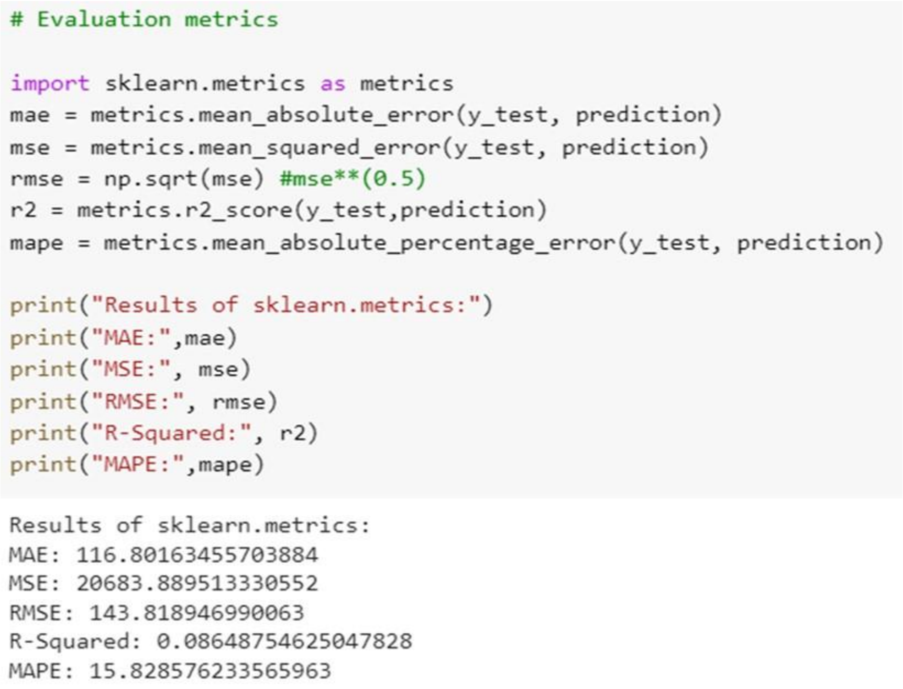
Fig 20. Evaluation metrics for Alcohol Frequency Use Linear Regression Model in the past year



```
# Evaluation metrics

import sklearn.metrics as metrics
mae = metrics.mean_absolute_error(y_test, prediction)
mse = metrics.mean_squared_error(y_test, prediction)
rmse = np.sqrt(mse) #mse**(0.5)
r2 = metrics.r2_score(y_test,prediction)
mape = metrics.mean_absolute_percentage_error(y_test, prediction)

print("Results of sklearn.metrics:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
print("MAPE:",mape)
```

```
Results of sklearn.metrics:
MAE: 67.3325347936132
MSE: 7585.800442891922
RMSE: 87.09650074998376
R-Squared: 0.0773372152438615
MAPE: 6.050438707733569
```

Fig 21. Evaluation metrics for Drug Frequency Use Linear Regression Model in the past year

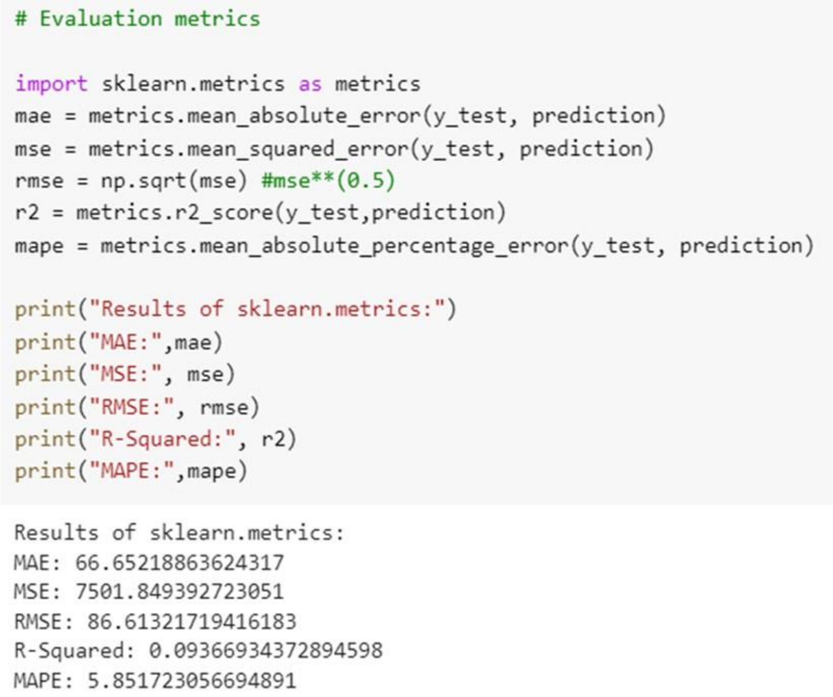


```
# Evaluation metrics

import sklearn.metrics as metrics
mae = metrics.mean_absolute_error(y_test, prediction)
mse = metrics.mean_squared_error(y_test, prediction)
rmse = np.sqrt(mse) #mse**(0.5)
r2 = metrics.r2_score(y_test,prediction)
mape = metrics.mean_absolute_percentage_error(y_test, prediction)

print("Results of sklearn.metrics:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
print("MAPE:",mape)
```

```
Results of sklearn.metrics:
MAE: 116.80163455703884
MSE: 20683.889513330552
RMSE: 143.818946990063
R-Squared: 0.08648754625047828
MAPE: 15.828576233565963
```

Random Forest Regressor Prior to implementing this model we also conduct hyperparameter tuning, like in the Random Forest classifier.

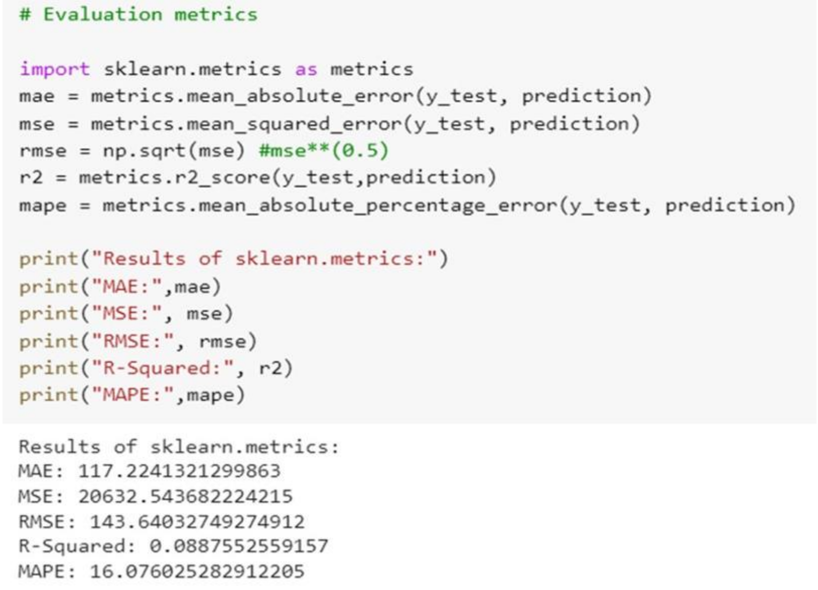
Fig 22. Evaluation metrics for Alcohol Frequency Use Random Forest Regression Model in the past year



```
# Evaluation metrics

import sklearn.metrics as metrics
mae = metrics.mean_absolute_error(y_test, prediction)
mse = metrics.mean_squared_error(y_test, prediction)
rmse = np.sqrt(mse) #mse**(0.5)
r2 = metrics.r2_score(y_test,prediction)
mape = metrics.mean_absolute_percentage_error(y_test, prediction)

print("Results of sklearn.metrics:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
print("MAPE:",mape)
```

```
Results of sklearn.metrics:
MAE: 66.65218863624317
MSE: 7501.849392723051
RMSE: 86.61321719416183
R-Squared: 0.09366934372894598
MAPE: 5.851723056694891
```

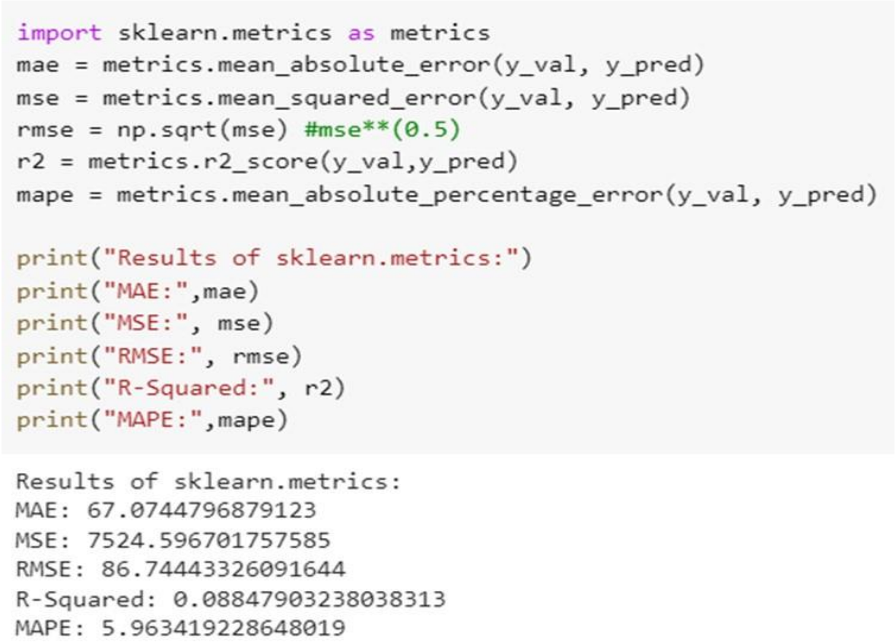
Fig 23. Evaluation metrics for Drug Frequency Use Random Forest Regression Model in the past year.



```
# Evaluation metrics

import sklearn.metrics as metrics
mae = metrics.mean_absolute_error(y_test, prediction)
mse = metrics.mean_squared_error(y_test, prediction)
rmse = np.sqrt(mse) #mse**(0.5)
r2 = metrics.r2_score(y_test,prediction)
mape = metrics.mean_absolute_percentage_error(y_test, prediction)

print("Results of sklearn.metrics:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
print("MAPE:",mape)
```

```
Results of sklearn.metrics:
MAE: 117.2241321299863
MSE: 20632.543682224215
RMSE: 143.64032749274912
R-Squared: 0.0887552559157
MAPE: 16.076025282912205
```

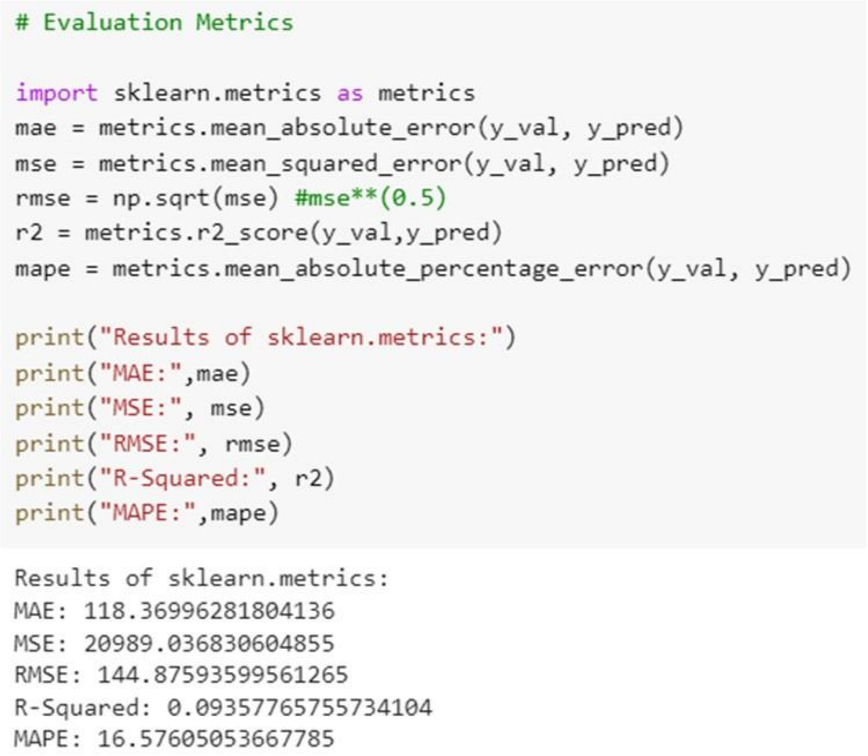
CatBoost Regressor Fig 24. Catboost Regressor Model for Alcohol use in the past year Evaluation Metrics



```
import sklearn.metrics as metrics
mae = metrics.mean_absolute_error(y_val, y_pred)
mse = metrics.mean_squared_error(y_val, y_pred)
rmse = np.sqrt(mse) #mse**(0.5)
r2 = metrics.r2_score(y_val,y_pred)
mape = metrics.mean_absolute_percentage_error(y_val, y_pred)

print("Results of sklearn.metrics:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
print("MAPE:",mape)
```

```
Results of sklearn.metrics:
MAE: 67.0744796879123
MSE: 7524.596701757585
RMSE: 86.74443326091644
R-Squared: 0.08847903238038313
MAPE: 5.963419228648019
```

Fig 25. Catboost Regressor Model for Drug use in the past year Evaluation Metrics

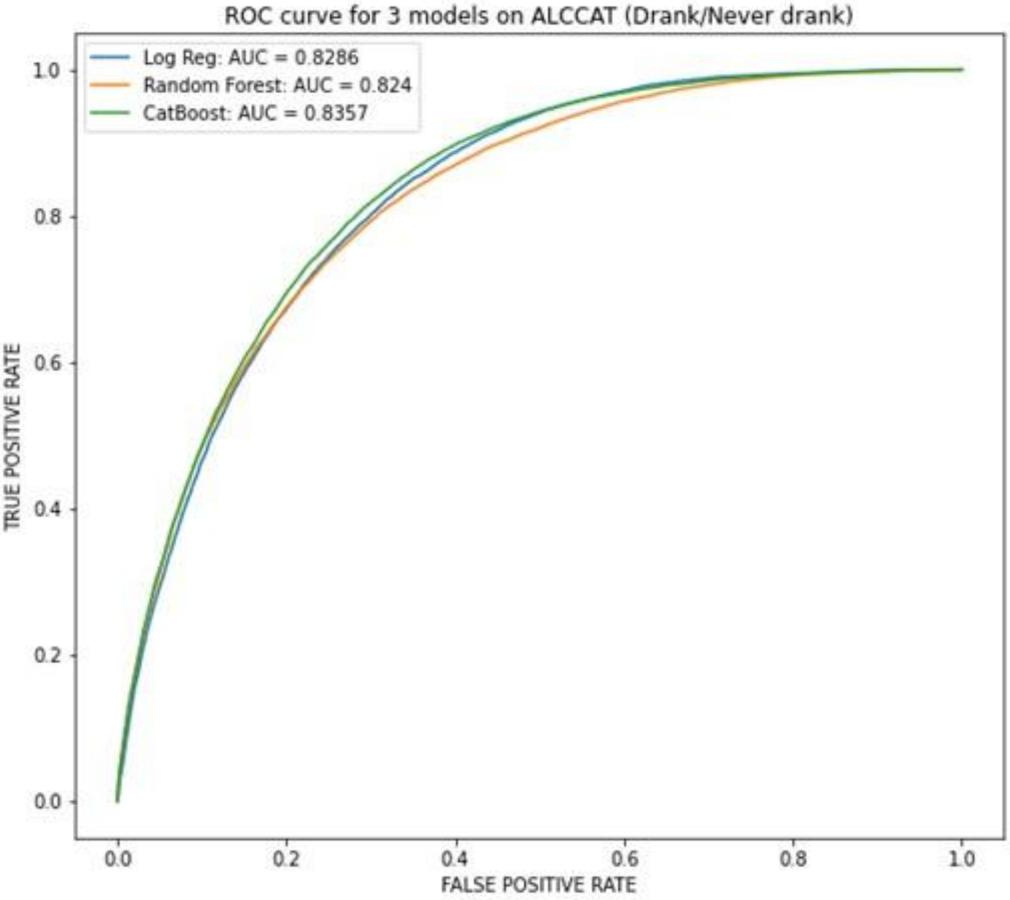


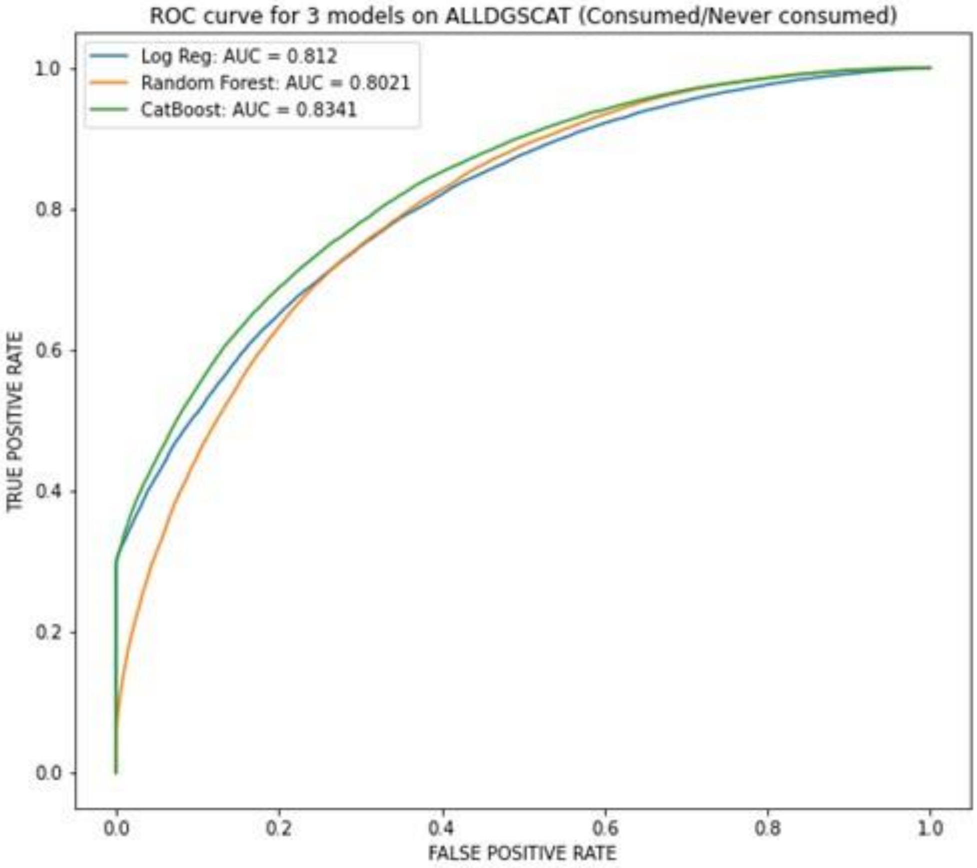
```
# Evaluation Metrics

import sklearn.metrics as metrics
mae = metrics.mean_absolute_error(y_val, y_pred)
mse = metrics.mean_squared_error(y_val, y_pred)
rmse = np.sqrt(mse) #mse**(0.5)
r2 = metrics.r2_score(y_val,y_pred)
mape = metrics.mean_absolute_percentage_error(y_val, y_pred)

print("Results of sklearn.metrics:")
print("MAE:",mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R-Squared:", r2)
print("MAPE:",mape)
```

```
Results of sklearn.metrics:
MAE: 118.36996281804136
MSE: 20989.036830604855
RMSE: 144.87593599561265
R-Squared: 0.09357765755734104
MAPE: 16.57605053667785
```

5.3 Model Evaluation

Alcohol use (categorical) in the past year The three models that were implemented are close in performance, however, the best performing model is the Catboost Model (as evidenced in figure 26).


ROC curve for 3 models on ALCCAT (Drank/Never drank)
Log Reg: AUC = 0.8286
Random Forest: AUC = 0.824
CatBoost: AUC = 0.8357
TRUE POSITIVE RATE
FALSE POSITIVE RATE
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0



Drug use (categorical) in the past year Again, the Catboost classifier model is slightly more suitable at predicting drug use.

Alcohol use in the past year (Frequency)			
Evaluation Metrics	Linear Regression	Random Forest	CatBoost
MAE	67.3325347936132	66.65218863624317	67.0744796879123
MSE	7585.800442891922	7501.849392723051	7524.596701757585
RMSE	87.09650074998376	86.61321719416183	86.74443326091644
R-Squared	0.0773372152438615	0.09366934372894598	0.08847903238038313
MAPE	6.050438707733569	5.851723056694891	5.963419228648019

Using root mean square error (RMSE) and mean absolute percentage error (MAPE) we can conclude that the best model is the random forest regression.

For predicting frequency of alcohol use.

Drug use in the past year (Frequency)			
Evaluation Metrics	Linear Regression	Random Forest	CatBoost
MAE	116.80163455703884	117.2241321299863	118.36996281804136
MSE	20683.889513330552	20632.543682224215	20989.036830604855
RMSE	143.818946990063	143.64032749274912	144.87593599561265
R-Squared	0.0864875462504782	0.0887552559157	0.09357765755734104
MAPE	15.828576233565963	16.076025282912205	16.57605053667785

Once again, the best performing model is the random forest regression, as evidenced by RMSE and MAPE.

6 Summary of findings

Both our regression models and our classification models showed how the factors we examined had a relationship to drug and alcohol use. This means that we can reject our null hypothesis: The examined factors have a relation to substance use in the past year. Furthermore, certain features had a higher on alcohol or drug use.

7 Limitations

While we are satisfied with our work, there were limitations to our project. The first limitation is that much of our dataset is categorical in nature, which made certain models difficult to implement. Another limitation is that our team is relatively new to implementing machine learning models. While we consider our performance to be satisfactory, our team is not as effective in implementing these models, which could have possibly hindered further steps in our research process. Our final limitation is our lack of domain specific knowledge. None of our team members are experts on substance abuse. Due to this, our research questions were formulated based on previous literature. If we had more knowledge about the topic we might have formulated different questions.

8 Appendix:

*Provided in the Code file
https://colab.research.google.com/drive/1lOGuNsDGlbwOrrRWIoYu3V6ZO3dQRwHg?usp=sharing

References

1. [i] Volkow, N. D., Poznyak, V., Saxena, S., Gerra, G. (2017). substance use disorders: impact of a public health rather than a criminal justice approach. World Psychiatry, 16(2), 213–214. https://doi.org/10.1002/wps.20428

2. [ii] NIDA. 2020, May 25. What are risk factors and protective factors?. Retrieved from https://nida.nih.gov/publications/preventing-drug-use-among-children-adolescents/chapter-1-risk-factors-protective-factors/what-are-risk-factors, 2022, September 27

3. [iii] Center for Behavioral Health Statistics and Quality. (2021). 2020 National Survey on substance Use and Health Public Use File Codebook, Sub- stance Abuse and Mental Health Services Administration, Rockville, MD https://www.samhsa.gov/data/data-we-collect/nsduh-national-survey-substance-use-and-health Center for Disease Control and Prevention. (2022). High Risk Substance Use in Youth. https://www.cdc.gov/healthyyouth/substance-use/index.htm NIDA. 2020, May 25. What are risk factors and protective factors?. Retrieved from https://nida.nih.gov/publications/preventing-substance-use-among-children-adolescents/chapter-1-risk-factors-protective-factors/what-are-risk-factors,
2022,
September 27

4. [iv] Rosner, B., Neicun, J., Yang, J. C., Roman-Urrestarazu, A. (2021). Substance use among sexual minorities in the US – Linked to inequalities and unmet need for mental health treatment? Results from the National Survey on sub- stance Use and Health (NSDUH). Journal of Psychiatric Research, 135, 107–118. https://doi.org/10.1016/j.jpsychires.2020.12.023

5. [v] Waddell, J. T. (2021). Between- and within-group effects of alcohol and cannabis co- use on AUD/CUD in the NSDUH 2002–2019. substance and Alcohol Dependence, 225, 108768. https://doi.org/10.1016/j.substancealcdep.2021.108768

6. [vi] Lin, L. A., Ilgen, M. A., Jannausch, M., Bohnert, K. M. (2016). Com- paring adults who use cannabis medically with those who use recreation- ally: Results from a national sample. Addictive Behaviors, 61, 99–103. https://doi.org/10.1016/j.addbeh.2016.05.

7. [vii] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.