Drug Use Predictor: Predictive Modeling for Youth Drug Use

Abstract:

The undertaking articulates how, applying decision tree and random forest classification models, it is possible to forecast the usage of drugs by young people through the information presented. The objective is to carry out an assessment of causes linked to drug usage among youth and discrimination by two classifiers according to users and non-users of drugs. Other drug-related variables, such as the age at which alcohol consumption began and the frequency of smoking monthly, were found to be the best indicators for the models, while demographic and socioeconomic factors did not show a strong correlation with the response variables. The study's variables and modeling strategy, taken together, appear to have worked well for categorization purposes, but not for regression analysis aimed at predicting the smoking behavior of young people.

Introduction:

The widespread of youth drug use is a huge challenge that the status quo desperately wants to be addressed and the predictive models are capable of singling out potential users. This scenario is based on machine learning algorithms that examine the survey data and forecast youth drug use, employing significant qualitative and experiential factors. Using decision tree models applied to survey data from the National Survey on Drug Use and Health (NSDUH), this paper attempts to explore the relationship between juvenile drug use and several characteristics. The survey encompasses both household and non-institutional group quarter populations, covering a broad spectrum of demographics. The NSDUH 2020 dataset, which contains comprehensive data on respondents' demographics, experiences as youth, and drug use habits, is the source of the data for this research. The report uses a variety of decision tree modeling techniques, such as regression to predict the number of days a person will drink alcohol, binary classification to identify factors linked to cigarette use, and multiclass classification to distinguish between infrequent, occasional, and frequent marijuana use.

Theoretical Background:

Decision Tree: A decision tree is a kind of hierarchical tree structure that divides data into binary categories according to characteristics. Features are represented by internal nodes, and their potential values are represented by branching.

Decision trees are lie tools used to sort things into groups or making predictions. These work by splitting up the stuff you are looking for at into smaller groups based on their respective characteristics. At every step, this picks which characteristic to focus on next in order to make the best split. Basically, it’s like setting up a messy room into well and organized room.

If we feel like there are a lot in number of unnecessary branches in the decision tree, we can avoid those with one simple technique called **'Pruning'.** This process helps to reduce the number of branches in the tree, prevents overfitting the data. This can be used while building the decision tree or even after the decision tree.

Random Forest: Random Forest is an ensemble learning method that builds multiple decision trees during training. Each tree is trained on a random subset of the training data and a random subset of the features. While prediction, the output of each tree is averaged or voted to obtain the final prediction and also this approach helps to reduce overfitting the data.

Bagging: This technique is used to reduce the variance of learning method by averaging multiple models trained on different subsets of the training data. Each model is trained separately and their predictors are combined by averaging or majority voting.

Boosting: Boosting is an ensemble learning method that combines multiple weak learners to create a strong learner. This method usually gives high accuracy than bagging

It is like teamwork where each person learns from the mistakes of the others. First, we start with a simple learner. Then, we focus more on the things that it didn't do well and try to improve on those for the next learner. This keeps going until we have a strong team of learners. Common boosting methods include AdaBoost and Gradient Boosting.

Tuning: This is the process used to make the model perform better. It is basically adjusting the settings of a model to make it work better for that respective task. We use different values or setting to check the model performance and select the best value or setting where the model performance is at peaks

Hyperparameters for Each Model:

Decision Tree :

- max_depth: Maximum depth of the decision tree.
- min_samples_split: Minimum number of samples required to split an internal node.
- min_samples_leaf: Minimum number of samples required to be at a leaf node.

Bagging :

- n_estimators: Number of base estimators (decision trees) in the ensemble.
- AdaBoost Classifier:
- n_estimators: Number of base estimators (weak learners) in the ensemble.
- learning_rate: Rate at which the contribution of each weak learner is weighted.

Random Forest Classifier:

- n_estimators: Number of decision trees in the random forest.
- max_depth: Maximum depth of each decision tree.

- min_samples_split: Minimum number of samples required to split an internal node.
- min_samples_leaf: Minimum number of samples required to be at a leaf node.

Gradient Boosting Regressor:

- n_estimators: Number of boosting stages (trees) to be used.
- learning_rate: Rate at which the contribution of each tree is weighted.
- max_depth: Maximum depth of each tree.
- min_samples_split: Minimum number of samples required to split an internal node.
- min_samples_leaf: Minimum number of samples required to be at a leaf node.

Methodology:

EDA : First step, exploring the dataset to understand its structure and distributions. This includes examining the columns, unique values in the and identifying missing values

Data Cleaning : This step was crucial especially in this task as there were 79 columns and 5500 rows of data. This step involves handling missing values using techniques like imputation, converting categorical data to factors.

Imputation : Used to handle the missing values in the data set. I have used mode of each columns is used to fill the missing values as the mean of the columns are mistake as there are some categorical variables.

Variable Name Replacing : To improve both readability and understandability, variable names are mapped to more understandable and descriptive names in the data set. This will improve to understand more about the output.

Models in Binary Classification: First, we picked the necessary variables about how youth behave and if they use drugs. Then divided the info into two parts one for training and the other for testing. It used teamwork methos like bagging and boosting to train decision makers each with different parameters to increase the performance. The best-performing model is determined based on cross-validated accuracy scores, and its feature importances are visualized to understand the influential factors in predicting tobacco use.

Models in Multi-Class Classification: First, we picked the necessary variables about how youth behave and if they use drugs. Splitted the dataset into training and testing of 20% and 80% and a random state of 42. Using 5-fold cross-validation, a Random Forest Classifier with 100 estimators is implemented and evaluated its mean accuracy score. Next, using the same cross-validation setup, AdaBoostClassifier models with different shrinkage values are trained and evaluated. The best model is determined by taking the mean accuracy value and selecting it. Ultimately, a horizontal bar plot is used to compute and display the feature importances of the best model, giving insights into the relative importance of each feature in the classification task.

Models in Regression: First, we picked the necessary variables about how youth behave and if they use drugs. Here the I considered the usage of cigarette days in month is predicted using a set of features related to student behaviour and demographics. Splitted the data into training and testing sets for further process. Here I have used Decision Tree regressor which computes the mean squared error(mse) with 5 fold cross validation and also Gradient Boosting Regressor models with varying shrinkage values. In order to provide insights into the predictive factors influencing the target variable, it finally selects the model with the lowest mean square error (MSE) that performs the best. It also visualizes the feature importances and shows the decision tree for the pruned Gradient Boosting Regressor.

Computational Results:

Binary Classification : Here this model is used to predict the likelihood of tobacco usage based on the various other predictor variables related to youth data and theoir demographics Here I have used two techniques to predict the need which are Bagging and Boosting
Model Performance :
Bagging Classifier:

Cross-validation scores ranged from approximately 0.8918 to 0.9127 across different folds, with a mean accuracy of approximately 0.9025.

AdaBoost Classifier:

Different shrinkage values were tested (0.1, 0.01, 0.001, 0.0001) to optimize the model's learning rate.

The best-performing model achieved a mean accuracy of approximately 0.9109 with a shrinkage value of 0.1.

Feature importance : I have plotted the horizontal bargraph to reveal the importance of the predictor variables on the target variable

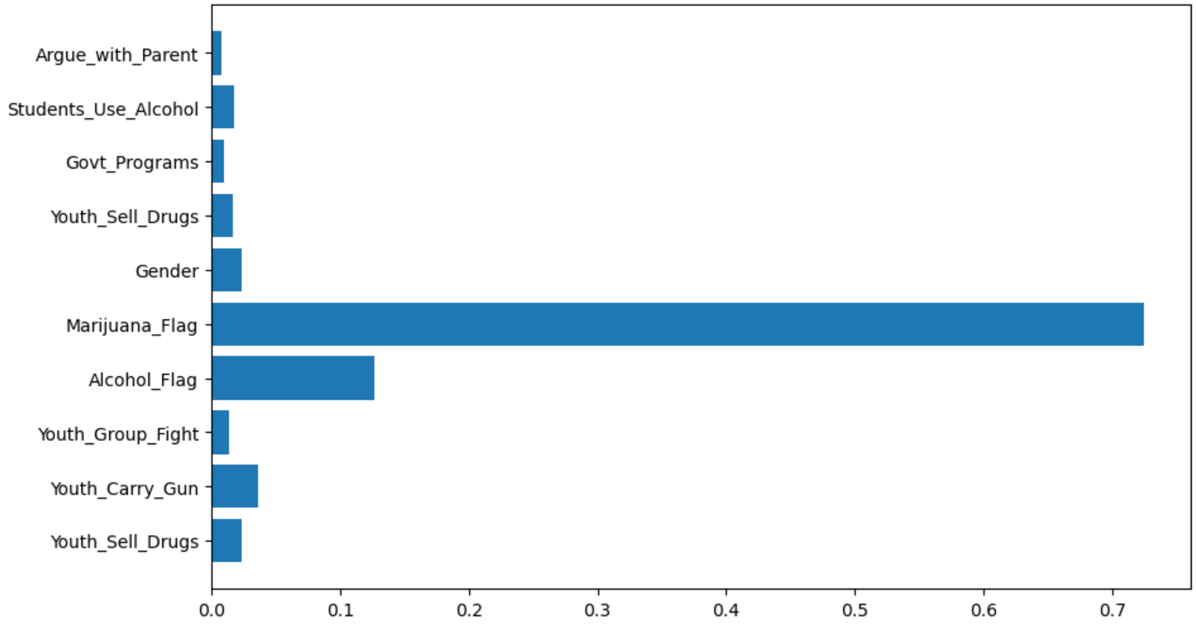

Figure 1 : Horizontal Bar Graph of Variable Importance – Binary Classification

Marijuana Flag: This feature exhibited the highest importance, with a value of 0.72(approx), indicating its strong predictive power in determining tobacco usage.

Alcohol Flag: The second-highest importance was attributed to the alcohol flag, with a value of 0.15(approx), suggesting its moderate influence on tobacco usage prediction.

Multi-class Classification : Here this model is used to predict the likelihood of cigarettes usage in days in past month based on the various other predictor variables related to youth data and their demographics.

Random Forest Cross-validation Scores:

At various data folds, the Random Forest model produced remarkable cross-validation scores that ranged from roughly 0.979 to 0.983.

The Random Forest model's mean cross-validation accuracy was found to be roughly 0.982, demonstrating strong accuracy and resilience.

Cross-validation scores for Gradient Boosting:

Gradient Boosting models with various shrinkage (learning rate) values were assessed.

The model consistently achieved high cross-validation scores, ranging from roughly 0.983 to 0.9835, across all tested shrinkage values (0.1, 0.01, 0.001, 0.0001).

Strong predictive performance is shown by the Gradient Boosting models mean accuracy, which is almost same at roughly 0.983.
Best model selection : Based on the above results the best model identifies was AdaBoostClassifier with a learning rate(Shrinkage values) of 0.001 and 100 estimators.

Coming to featiure importance :

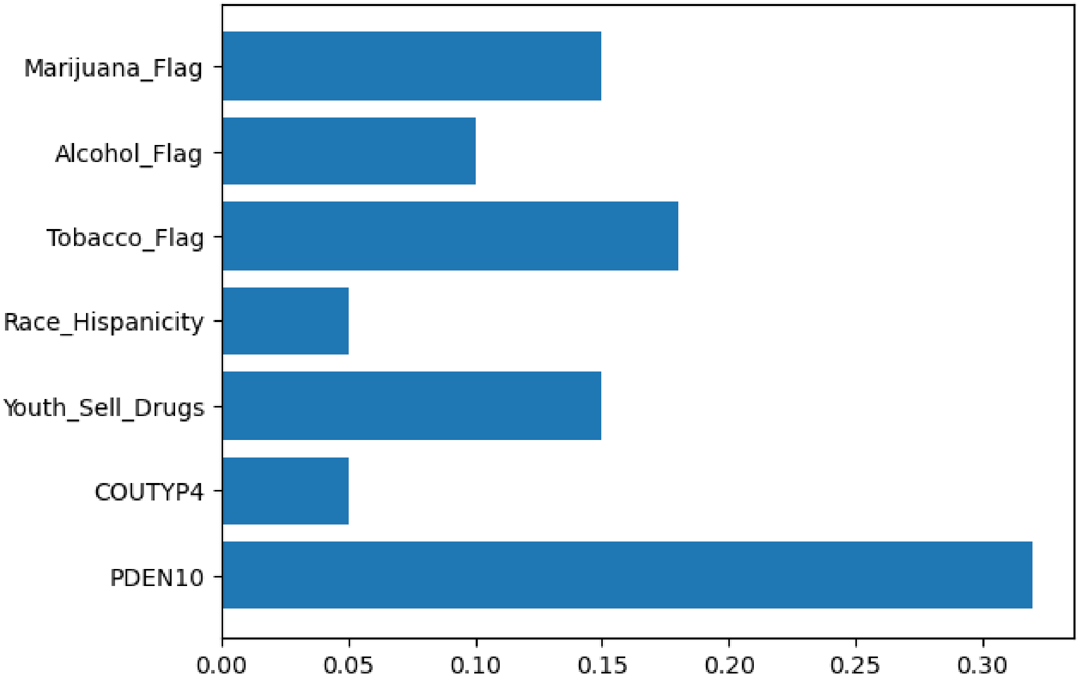

Figure 2 : Horizontal Bar Graph of Variable Importance – Multi-class Classification

With an importance score of 0.32, the variable Population density of 2010 was found to be the most significant predictor.

With scores of 0.17 and 0.15, respectively, the Tobacco Usage and Marijuana Flag variables closely followed and showed significant importance.

Regression : : Here this model is used to predict the likelihood of cigarettes usage in days in past month based on the various other predictor variables related to youth data and their demographics.

Decision Tree Cross-Validation MSE: With Decision Tree Regressor cross-validation, the mean squared error (MSE) is roughly 0.446.

Gradient Boosting Cross-Validation MSE : The mean square errors (MSE) derived from cross-validation with a gradient boosting regression for various shrinkage values are reported below:

MSE equals 0.245 for shrinkage value of 0.1.

MSE is 0.242 for a shrinkage value of 0.01.

MSE equals 0.249 for shrinkage value 0.001.

MSE equals 0.250 for shrinkage value 0.0001.
from the above computational results of 4 different shrinkage values. There is not much huge difference in the mse but if we take the minute changes into consideration then the shrinkage value of 0.01 gives the best mse so we can say that the best model is gradient boosting model with shrinkage value of 0.01

Variable Importance:

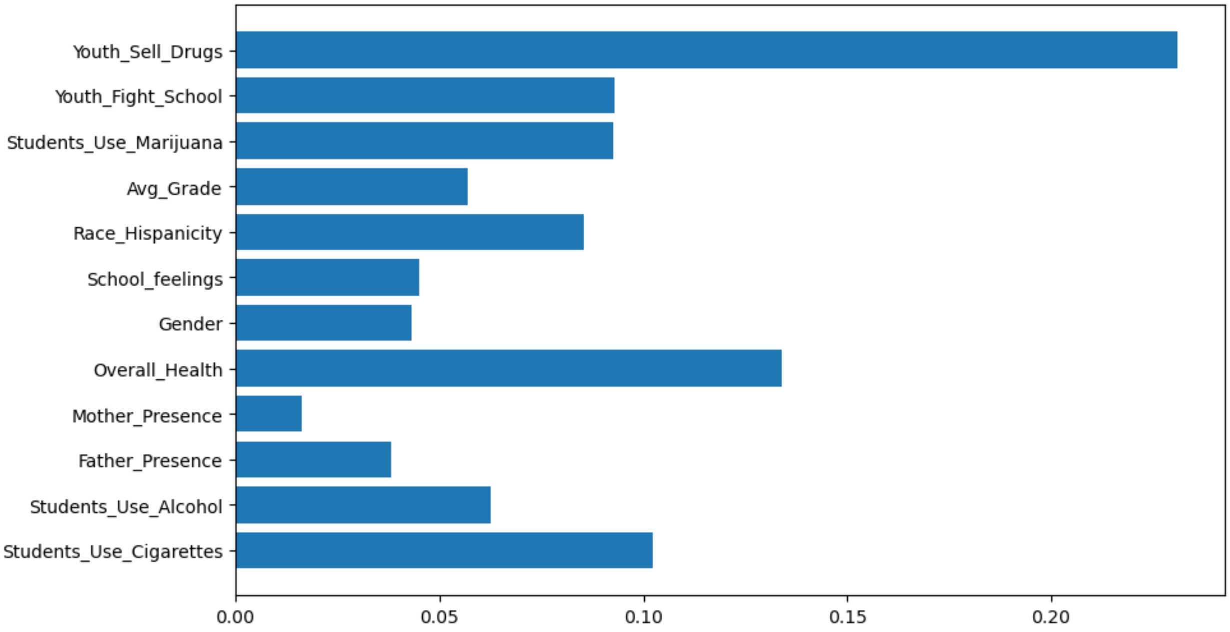

Figure 3 : Horizontal Bar Graph of Variable Importance – Regression

"Youth_Sell_Drugs" has an importance score of roughly 0.23, making it the most significant predictor for the target variable, according to the horizontal bar chart illustrating variable importance. With a score of roughly 0.14, "Overall_health" comes in second in importance after that.

Decision Tree Plot : (After pruning)

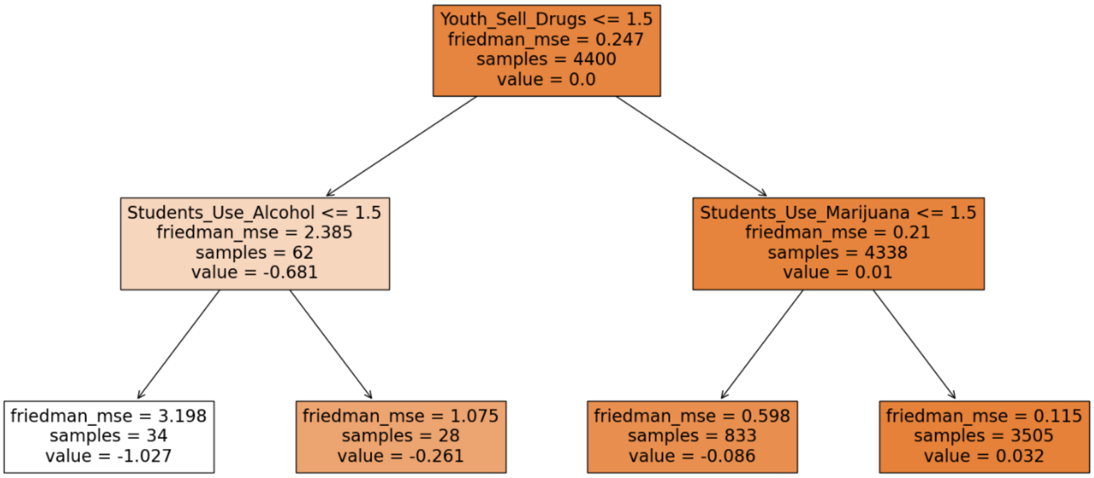

Figure 4 : Decision Tree – Regression

This tree illustrates the model's decision-making process by demonstrating how various features affect the expected result. The leaf nodes provide the predicted value based on the conditions met along the path from the root node, and each decision node represents a split on a feature.

Lets Breakdown the tree and see in-detail:

- ➔ The first split is based on the feature "Youth_Sell_Drugs." If this feature's value is less than or equal to 1.50, the model proceeds to the left branch, otherwise, it goes to the right branch.
- ➔ Within the left branch, further division is made based on the feature "Students_Use_Alcohol." If its value is less than or equal to 1.50, the model predicts a value of -1.03. Otherwise, it predicts a value of -0.26.
- ➔ If the value of "Youth_Sell_Drugs" is greater than 1.50, the model goes to the right branch.
- ➔ Within the right branch, the split is based on "Students_Use_Marijuana." If its value is less than or equal to 1.50, the model predicts a value of -0.09. Otherwise, it predicts a value of 0.03.

Discussion:

Binary Classificarion:

From the importance plot we can discuss that the most important factor in predicting the tobacco usage is marijuana usage followed by alcohol use. This explains that these behaviours are important in understanding tobacco use among young people.

Multi-Class Classification:

From the importance plot we can discuss that the most important factor in predicting the tobacco usage is how crowded an area was in 2010. This explains that the population is directly proportional to the usage of tobacco.

Regression:

From the importance plot we can discuss that the most important factor in predicting the how much someone smoke is whether selling drugs. This explains that there is direct strong relation between selling drugs and smoking.

The second factor which is Overall Health which indicates that the healthier the people the less they smoke. The unhealthier the people the more they smoke (Inversly Proportional).

Conclusions:

The results of the three studies showed that factors such as alcohol and marijuana use were significant predictors of smoking behavior. We also observed that population density has an impact. This indicates that a person's lifestyle and occupation have an impact on whether or not they smoke, particularly for young people.

This is crucial for developing policies and programs that encourage people to quit smoking and lead healthier lives. If we comprehend the reasons behind people's smoking, we can better design programs to assist them in quitting or preventing smoking altogether. Ensuring that youth can lead healthier lives free from tobacco use is the main goal.

Data from the large-scale NSDUH2020 survey, which asked youth about their lives and habits, were examined in our study. Growing up in a society where drinking and smoking are accepted can also encourage young people to use these substances, which is bad for their long-term health.

References:

1. National Survey on Drug Use and Health. (2020). [https://www.samhsa.gov/data/release/2020-national-survey-drug-use-and-health-nsduh-releases]

2. Scikit-learn Documentation. [https://scikit-learn.org/0.21/documentation.html]

Appendix Code:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor, plot_tree, export_text
from sklearn.impute import SimpleImputer
from sklearn.ensemble import GradientBoostingClassifier, GradientBoostingRegressor, RandomForestClassifier, RandomForestRegressor, BaggingClassifier, BaggingRegressor, AdaBoostClassifier
from sklearn.metrics import accuracy_score, classification_report, mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score

path = "youth_data.csv"
df = pd.read_csv(path)
print(df.columns)

for column in df.columns:
    unique_values = df[column].unique()
    print(f"'{column}': {unique_values}")

if df.isna().any().any():
    print("Yes")
else:
    print("No")

def impute(df):
    return df.mode().iloc[0]
for col in df.columns:
    imp_vals=impute(df[col])
    df[col].fillna(imp_vals,inplace=True)
```

```
n_after=df.isnull().sum()
imputed_columns=sum(n_after>0)
print(imputed_columns)

if df.isna().any().any():
    print("Yes")
else:
    print("No")

column_mapping = {
    "alcmdays" : "Alcohol_Days_Month",
    "mrjmdays" : "Marijuana_Days_Month",
    "cigmdays" : "Cigarette_Days_Month",
    "alcydays" : "Alcohol_Days_Year",
    "mrjydays" : "Marijuana_Days_Year",
    "mrjflag" : "Marijuana_Flag",
    'tobflag': 'Tobacco_Flag',
    'alcflag': 'Alcohol_Flag',
    "ircigfm" : "Cigarette_Frequency_Month",
    "irsmklsstry" : "Smokeless_Tobacco_Frequency_Month",
    "iralcage" : "Alcohol_Age_First_Use",
    "irmjage" : "Marijuana_Age_First_Use",
    "irmjfm" : "Marijuana_Frequency_Month",
    "iralcfm" : "Alcohol_Frequency_Month",
    'iralcfy':'Alcohol_frequency',
    'irmjfy':'Marijuana_freuency',
    "NEWRACE2" : "Race_Hispanicity",
    "irsex" : "Gender",
    "HEALTH2" : "Overall_Health",
    'imother': 'Mother_Presence',
    'ifather': 'Father_Presence',
    'income': 'Income_Level',
    'tchgjob': 'Teacher_Job_Performance',
    'stnddnk': 'Drinking_Standard',
     "argupar" : "Argue_with_Parent",
    "YOGRPFT2" : "Youth_Group_Fight",
    "YOFIGHT2" : "Youth_Fight_School",
    "YOHGUN2" : "Youth_Carry_Gun",
    "YOSELL2" : "Youth_Sell_Drugs",
    "govtprog" : "Govt_Programs",
    "avggrade" : "Avg_Grade",
    "stndscig" : "Students_Use_Cigarettes",
    "stndalc" : "Students_Use_Alcohol",
    "stndsmj" : "Students_Use_Marijuana",
    "schfelt" : "School_feelings"

}
```

```
df.rename(columns=column_mapping, inplace=True)
df.columns

for column in df.columns:
    unique_values = df[column].unique()
    print(f"'{column}': {unique_values}")
```

# BINARY CLASSIFICATION

```
bin_cl =
['Youth_Sell_Drugs','Youth_Carry_Gun','Youth_Group_Fight','Alcohol_Flag','Marijuana_Flag','Gender','Y
outh_Sell_Drugs', 'Govt_Programs','Students_Use_Alcohol','Argue_with_Parent']

X = df[bin_cl]
y = df['Tobacco_Flag']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

bagging_fit_bin = BaggingClassifier( n_estimators=100, random_state=42)

bagging_fit_bin .fit(X_train,y_train)
bagging_cv_score_bin = cross_val_score(bagging_fit_bin , X, y, cv=5)
print("Bagging Cross-validation scores:", bagging_cv_score_bin)
print("Mean Cross-validation accuracy - Bagging:", bagging_cv_score_bin.mean())

shrinkage_values = [0.1, 0.01,0.001, 0.0001]
best_model = None
best_acc = 0
for shrinkage in shrinkage_values:
    boosting_model_bin = AdaBoostClassifier(n_estimators=100, learning_rate=shrinkage,
random_state=42)
    boosting_model_bin.fit(X_train,y_train)
    cv_scores = cross_val_score(boosting_model_bin, X, y, cv=5)
    mean_acc = cv_scores.mean()
    print("Shrinkage :", shrinkage)
    print("Cross-Validation Scores:", cv_scores)
    print("Mean Accuracy:", mean_acc)
    print()
    if mean_acc > best_acc:
        best_acc = mean_acc
        best_model = boosting_model_bin
print("Best Model:")
print(best_model)
print("Best Accuracy:", best_acc)

boosting_model_bin.fit(X_train,y_train)
```

```
importances=boosting_model_bin.feature_importances_
plt.figure(figsize=(10,6))
plt.barh(range(len(bin_cl)),importances,align="center")
plt.yticks(range(len(bin_cl)),bin_cl)
```

# MULTI-CLASSIFICATION

```
mul_cl =
['PDEN10','COUTYP4','Youth_Sell_Drugs','Race_Hispanicity','Tobacco_Flag','Alcohol_Flag','Marijuana_Fla
g']
X = df[mul_cl ]
y = df['Cigarette_Days_Month']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf_fit_mul = RandomForestClassifier(n_estimators=100, random_state=42)
rf_score_mul = cross_val_score(rf_fit_mul, X, y, cv=5)
print("Random Forest Cross-validation scores:", rf_score_mul)
print("Mean Cross-validation accuracy -Random Forest:", rf_score_mul.mean())

shrinkage_values = [0.1, 0.01,0.001, 0.0001]
best_fit_mul = None
best_acc_mul = 0
for shrinkage in shrinkage_values:
   boosting_model = AdaBoostClassifier(n_estimators=100, learning_rate=shrinkage, random_state=42)
   cv_scores = cross_val_score(boosting_model, X, y, cv=5)
   mean_acc_mul = cv_scores.mean()
   print("Shrinkage (Learning Rate):", shrinkage)
   print("Cross-Validation Scores:", cv_scores)
   print("Mean Accuracy:", mean_acc_mul)
   print()
   if mean_acc_mul > best_acc_mul:
      best_acc_mul = mean_acc_mul
      best_fit_mul = boosting_model
print("Best Model:")
print(best_fit_mul)
print("Best Accuracy:", best_acc_mul)

best_fit_mul.fit(X,y)
importances=best_fit_mul.feature_importances_
plt.barh(mul_cl ,importances)
```

# REGRESSION

```
reg_cl  = ['Students_Use_Cigarettes','Students_Use_Alcohol','Father_Presence',
'Mother_Presence', 'Overall_Health', 'Gender', 'School_feelings',
'Race_Hispanicity','Avg_Grade','Students_Use_Marijuana','Youth_Fight_School','Youth_Sell_Drugs']

X = df[reg_cl]

y = df['Cigarette_Days_Month']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

dt_reg = DecisionTreeRegressor(random_state=42)
dt_cv_scores = cross_val_score(dt_reg, X, y, cv=5, scoring='neg_mean_squared_error')

print("Decision Tree Cross-Validation MSE:", -dt_cv_scores.mean())

shrinkage_values = [0.1, 0.01,0.001, 0.0001]

for i in shrinkage_values:
   gb_reg = GradientBoostingRegressor(n_estimators=100, learning_rate=i, random_state=42)
   gb_cv_scores = cross_val_score(gb_reg, X, y, cv=5, scoring='neg_mean_squared_error')
   print("Gradient Boosting Cross-Validation MSE:", -gb_cv_scores.mean(),'for shrinkage value',i)

mse_values = {
   "Gradient Boosting": min(-gb_cv_scores),
   "Decision Tree": min(-dt_cv_scores)
}

best_model = min(mse_values, key=mse_values.get)
best_mse = mse_values[best_model]

print("The Best Model is:", best_model)
print("MSE of the Best Model is:", best_mse)

model=GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, random_state=42)
model.fit(X,y)
importances=model.feature_importances_
plt.figure(figsize=(10,6))
plt.barh(range(len(reg_cl)),importances,align="center")
plt.yticks(range(len(reg_cl)),reg_cl)

plt.figure(figsize=(20, 10))
plot_tree(model.estimators_[0][0], feature_names=reg_cl , filled=True)
plt.show()
```

```
gb_reg_prune = GradientBoostingRegressor(n_estimators=100,
                                         max_depth=2,
                                         min_samples_split=4,
                                         min_samples_leaf=6,
                                         random_state=42)
gb_reg_prune.fit(X_train, y_train)
plt.figure(figsize=(20, 10))
plot_tree(gb_reg_prune.estimators_[0][0], feature_names=reg_cl, filled=True)
plt.show()
```

Thanks
Nikhil Raj Uppari