DRUG CONSUMPTION

<u>MEMBERS</u>

ABIN THOMAS

GEETHU SAMEELA

TOPIC TO COVER

Introduction → Problem overview → Dataset Overview → Data Cleaning and Preparation

→ Data Visualization → modelling → Model Evaluation → conclusion

Intodoction

- Drug consumption and addiction constitute a serious problem globally.
- The problem of evaluating an individual's risk of drug consumption and misuse is highly important.
- The problem of evaluating an individual's risk of drug consumption and misuse is highly important.
- our goal is to test if predicting drug consumption is possible and to identify the most informative attributes using data mining methods

Problem overview

- The problem statement involves a comprehensive analysis of drug consumption patterns using the 'drug consumption' dataset. This dataset encompasses various variables, including demographic information, socio-economic factors, and psychological characteristics, contributing to a multidimensional understanding of drug use behavior.

Dataset Overview

Key Features and Variables:

- Country: The country where the participant resides.
- Ethnicity: Ethnic background of the participant.
- Nscore, Escore, Oscore, Ascore, Cscore: Personality scores based on the NEO-PI-R personality inventory.
- Impulsive: Participant's impulsivity score.
- SS (Sensation Seeking): Sensation-seeking trait of the participant.
- Amphet, Amyl, Benzos, Caff, Cannabis, Choc, Coke, Crack, Ecstasy, Heroin, Ketamine, Legalh, LSD, Meth, Mushrooms, Nicotine, VSA: Binary variables indicating the consumption of different drugs.

Target Variable:

Semer: Binary variable indicating the participant's response regarding the consumption of the fictional drug "Semeron"

CL0: Never Used

CL1: Used over a Decade Ago

CL2: Used in the Last Decade

CL3: Used in the Last Year

CL4: Used in the Last Month

CL5: Used in the Last Week

CL6: Used in the Last Day.

Sample data

	ID	Age	Gender	Education	Country	Ethnicity	Nscore	Escore	Oscore	Ascore	...
0	1	0.49788	0.48246	-0.05921	0.96082	0.12600	0.31287	-0.57545	-0.58331	-0.91699	...
1	2	-0.07854	-0.48246	1.98437	0.96082	-0.31685	-0.67825	1.93886	1.43533	0.76096	...
2	3	0.49788	-0.48246	-0.05921	0.96082	-0.31685	-0.46725	0.80523	-0.84732	-1.62090	...
3	4	-0.95197	0.48246	1.16365	0.96082	-0.31685	-0.14882	-0.80615	-0.01928	0.59042	...
4	5	0.49788	0.48246	1.98437	0.96082	-0.31685	0.73545	-1.63340	-0.45174	-0.30172	...
...	...	...	...	...	...	...	...	...	...	...	...
95	96	1.09449	-0.48246	1.16365	0.96082	-0.31685	-0.46725	0.32197	1.06238	-1.62090	...
96	97	0.49788	0.48246	0.45468	0.24923	-0.31685	0.13606	-1.09207	0.29338	-0.15487	...
97	98	1.09449	0.48246	1.16365	0.96082	-0.31685	-0.46725	0.63779	-0.31776	-0.60633	...
98	99	0.49788	-0.48246	1.98437	0.24923	-0.31685	-1.19430	-0.43999	-0.01928	-0.60633	...
99	100	-0.07854	0.48246	0.45468	0.96082	-0.31685	-0.67825	3.00537	0.72330	0.94156	...

.	Ecstasy	Heroin	Ketamine	Legalh	LSD	Meth	Mushrooms	Nicotine	Semer	VSA
.	CL0	CL0	CL0	CL0	CL0	CL0	CL0	CL2	CL0	CL0
.	CL4	CL0	CL2	CL0	CL2	CL3	CL0	CL4	CL0	CL0
.	CL0	CL0	CL0	CL0	CL0	CL0	CL1	CL0	CL0	CL0
.	CL0	CL0	CL2	CL0	CL0	CL0	CL0	CL2	CL0	CL0
.	CL1	CL0	CL0	CL1	CL0	CL0	CL2	CL2	CL0	CL0
.	...	...	...	...	...	...	...	...	...	...
.	CL0	CL0	CL0	CL0	CL1	CL0	CL1	CL1	CL0	CL0
.	CL0	CL0	CL0	CL0	CL1	CL0	CL1	CL3	CL0	CL0
.	CL0	CL0	CL0	CL0	CL0	CL0	CL0	CL2	CL0	CL0
.	CL0	CL0	CL0	CL0	CL1	CL0	CL1	CL2	CL0	CL0
.	CL1	CL0	CL0	CL0	CL0	CL0	CL0	CL5	CL0	CL0

Data Cleaning and Preparation

- Handling Missing Values

```
data.isna().sum().sum()
```

```
0
```

- Encoding Categorical Variable

	Age	Gender	Education	Country	Ethnicity	Nscore	Escore	Oscore
0	0.49788	0.48246	-0.05921	0.96082	0.12600	0.31287	-0.57545	-0.58331
1	-0.07854	-0.48246	1.98437	0.96082	-0.31685	-0.67825	1.93886	1.43533
2	0.49788	-0.48246	-0.05921	0.96082	-0.31685	-0.46725	0.80523	-0.84732
3	-0.95197	0.48246	1.16365	0.96082	-0.31685	-0.14882	-0.80615	-0.01928
4	0.49788	0.48246	1.98437	0.96082	-0.31685	0.73545	-1.63340	-0.45174

	Ascore	Cscore	...	Crack	Ecstasy	Heroin	Ketamine	Legalh	LSD	Meth
0	-0.91699	-0.00665	...	0	0	0	0	0	0	0
1	0.76096	-0.14277	...	0	4	0	2	0	2	3
2	-1.62090	-1.01450	...	0	0	0	0	0	0	0
3	0.59042	0.58489	...	0	0	0	2	0	0	0
4	-0.30172	1.30612	...	0	1	0	0	1	0	0

	Mushrooms	Nicotine	VSA
0	0	2	0
1	0	4	0
2	1	0	0
3	0	2	0
4	2	2	0

[5 rows x 29 columns]

Conti…

- Normalization of Numerical Features

```python
# Display the dataset after normalization
print("\nNormalized and Imputed Dataset:")
print(X_imputed.head())
```

```
Normalized and Imputed Dataset:
        Age    Nscore    Escore    Oscore    Ascore    Cscore  Impulsive  \
0  0.527566  0.313500 -0.576912 -0.585137 -0.919341 -0.006281  -0.235108
1 -0.128854 -0.679764  1.944499  1.441683  0.763361 -0.142775  -0.752976
2  0.527566 -0.468308  0.807667 -0.850217 -1.625245 -1.016902  -1.453650
3 -1.123504 -0.149189 -0.808263 -0.018822  0.592338  0.586885  -1.453650
4  0.527566  0.736994 -1.637850 -0.453034 -0.302329  1.310098  -0.235108

         SS  Gender_-0.48246  Gender_0.48246  ...  Country_0.21128  \
0 -1.222226              0.0             1.0  ...              0.0
1 -0.220519              1.0             0.0  ...              0.0
2  0.420129              1.0             0.0  ...              0.0
3 -1.222226              0.0             1.0  ...              0.0
```

```
  Country_0.24923  Country_0.96082  Ethnicity_-1.10702  Ethnicity_-0.50212  \
              0.0              1.0                 0.0                 0.0
              0.0              1.0                 0.0                 0.0
              0.0              1.0                 0.0                 0.0
              0.0              1.0                 0.0                 0.0
              0.0              1.0                 0.0                 0.0

     city_-0.31685  Ethnicity_-0.22166  Ethnicity_0.1144  Ethnicity_0.126  \
               0.0                 0.0               0.0              1.0
               1.0                 0.0               0.0              0.0
               1.0                 0.0               0.0              0.0
               1.0                 0.0               0.0              0.0
               1.0                 0.0               0.0              0.0

  Ethnicity_1.90725
                0.0
                0.0
                0.0
                0.0
```

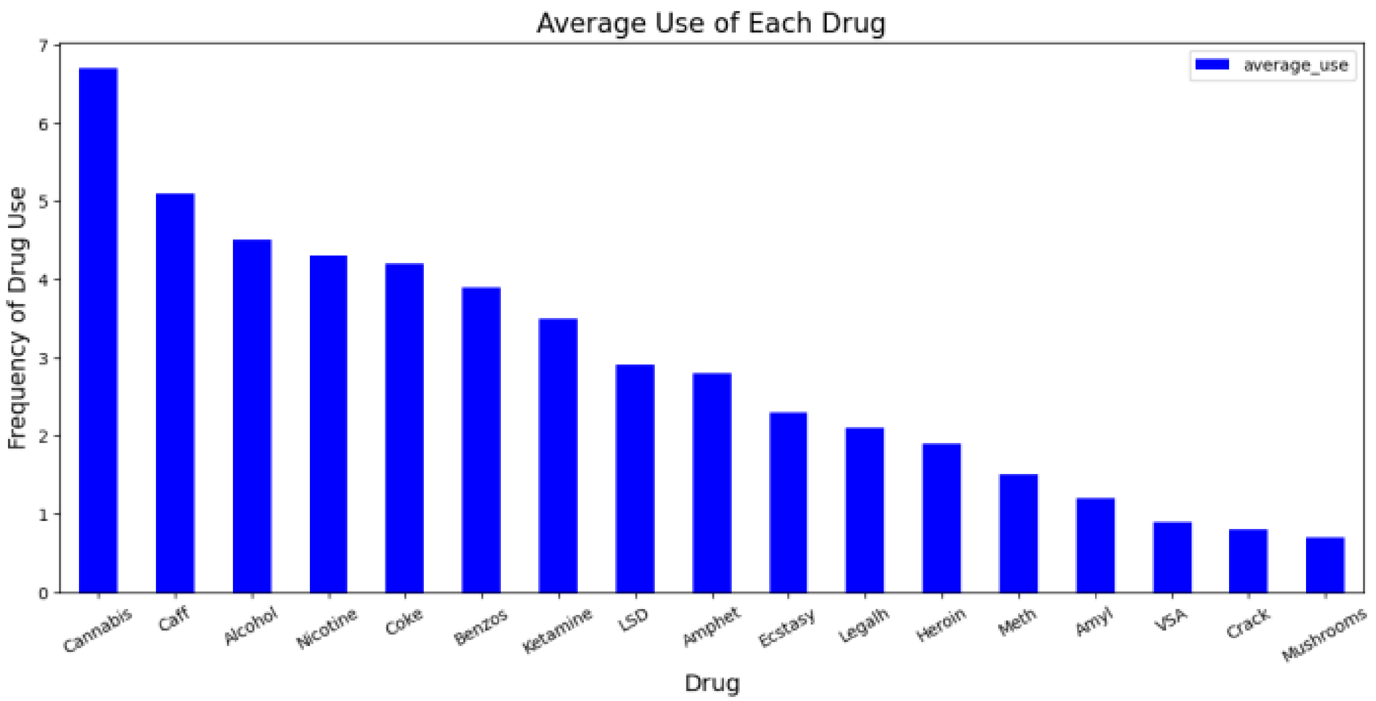
Cont..

Selection for each target variable

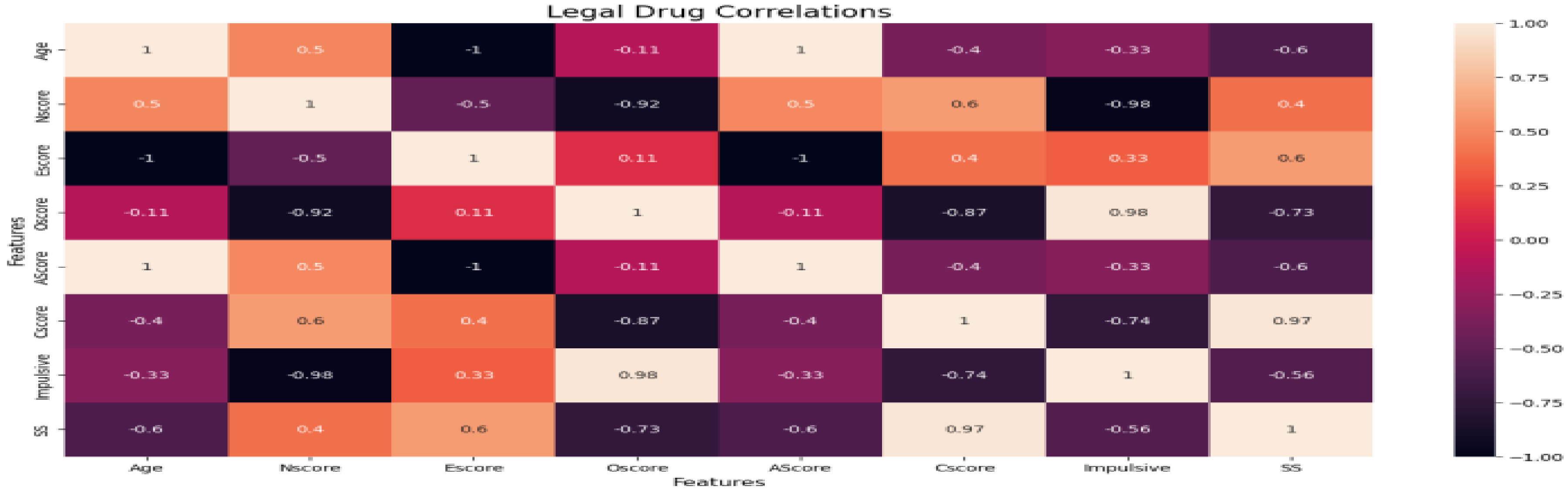
```python
# Feature selection for each target variable
k_best = 10
selected_features_per_target = {}

for target_column in y.columns:
    selector = SelectKBest(f_classif, k=k_best)
    X_selected = selector.fit_transform(X_imputed, y[target_column])

    selected_indices = selector.get_support(indices=True)

    selected_features = X_imputed.iloc[:, selected_indices]
    selected_features_per_target[target_column] = selected_features
for target_column, features in selected_features_per_target.items():
    print(f"Selected Features for {target_column}:")
    print(features.head())
```

```
Selected Features for Amphet:
        Age    Oscore    Cscore  Impulsive        SS  Gender_-0.48246  \
0  0.527566 -0.585137 -0.006281  -0.235108 -1.222226              0.0
1 -0.128854  1.441683 -0.142775  -0.752976 -0.220519              1.0
2  0.527566 -0.850217 -1.016902  -1.453650  0.420129              1.0
3 -1.123504 -0.018822  0.586885  -1.453650 -1.222226              0.0
4  0.527566 -0.453034  1.310098  -0.235108 -0.220519              0.0

   Gender_0.48246  Education_-0.61113  Country_-0.57009  Country_0.96082
0             1.0                 0.0               0.0              1.0
1             0.0                 0.0               0.0              1.0
2             0.0                 0.0               0.0              1.0
3             1.0                 0.0               0.0              1.0
4             1.0                 0.0               0.0              1.0
Selected Features for Amyl:
        Age    Cscore  Impulsive        SS  Gender_-0.48246  Gender_0.48246  \
0  0.527566 -0.006281  -0.235108 -1.222226              0.0             1.0
1 -0.128854 -0.142775  -0.752976 -0.220519              1.0             0.0
2  0.527566 -1.016902  -1.453650  0.420129              1.0             0.0
```

Data Visualization



heatmap




```
df.head()
```

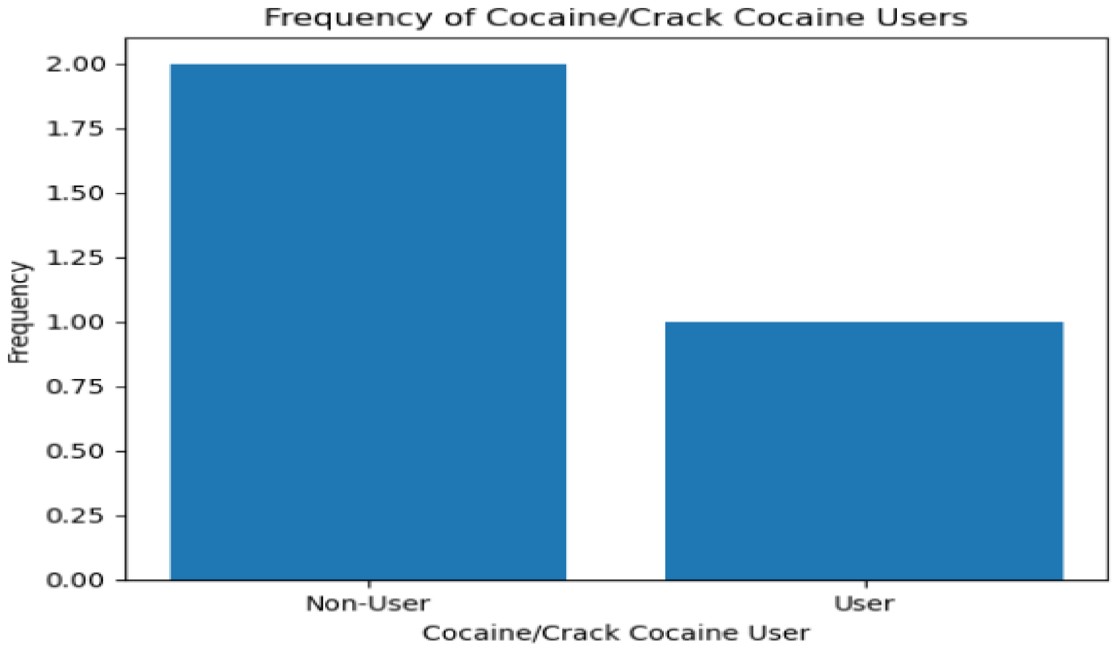
```
      drug  average_use
0  Alcohol          4.5
1     Amyl          1.2
2   Amphet          2.8
3   Benzos          3.9
4     Caff          5.1
```

Data preprocessing

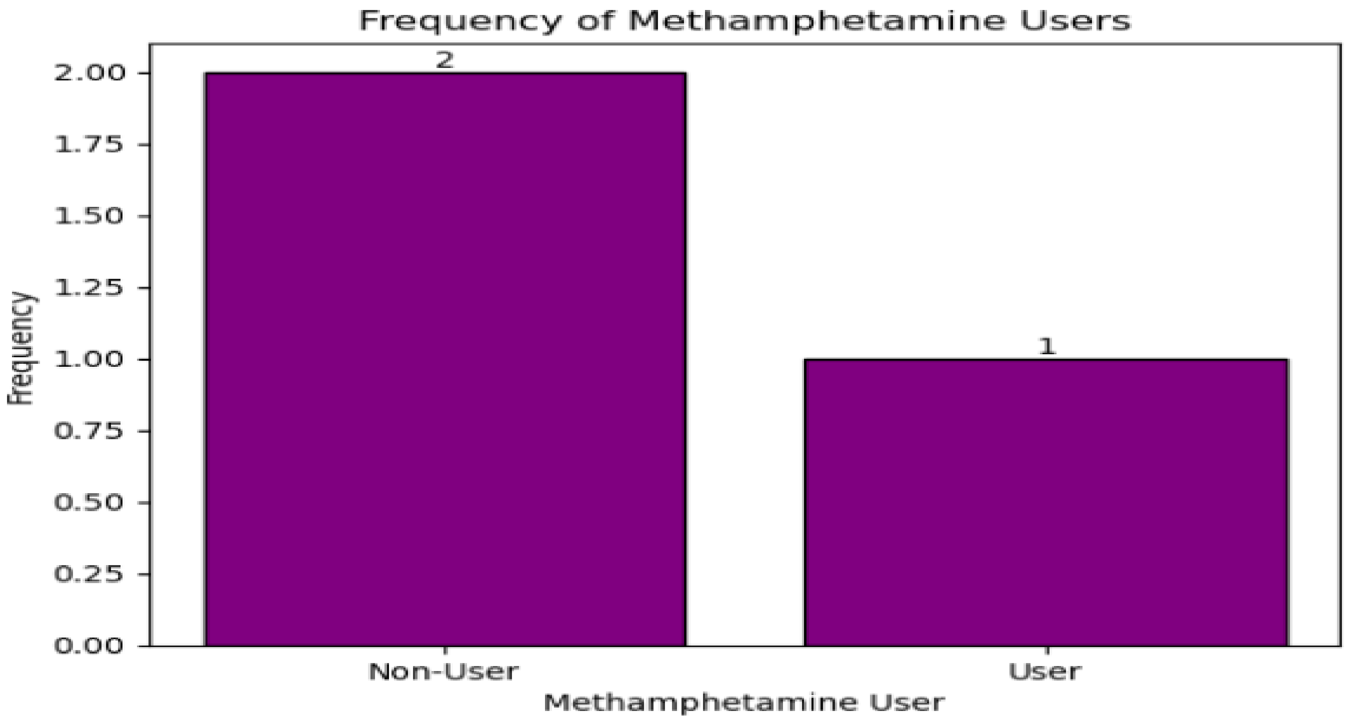
```
def preprocessing_inputs(df, column):
    df = df.copy()

    # Split df into X and y
    y = df[column]
    X = df.drop(column, axis=1)

    # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)

    # Scale X
    scaler = StandardScaler()
    scaler.fit(X_train)

    X_train = pd.DataFrame(scaler.transform(X_train),
                           index=X_train.index,
                           columns=X_train.columns)
    X_test = pd.DataFrame(scaler.transform(X_test),
                          index=X_test.index,
                          columns=X_test.columns)

    return X_train, X_test, y_train, y_test
```

```
def plot_confusion_matrix(y,y_predict):
    #Function to easily plot confusion matrix
    cm = confusion_matrix(y, y_predict)
    ax= plt.subplot()
    sns.heatmap(cm, annot=True, ax = ax, fmt='g', cmap='Blues');
    ax.set_xlabel('Predicted labels')
    ax.set_ylabel('True labels')
    ax.set_title('Confusion Matrix');
    ax.xaxis.set_ticklabels(['non-user', 'user']); ax.yaxis.set_ticklabels(['non-user', 'user'])
```

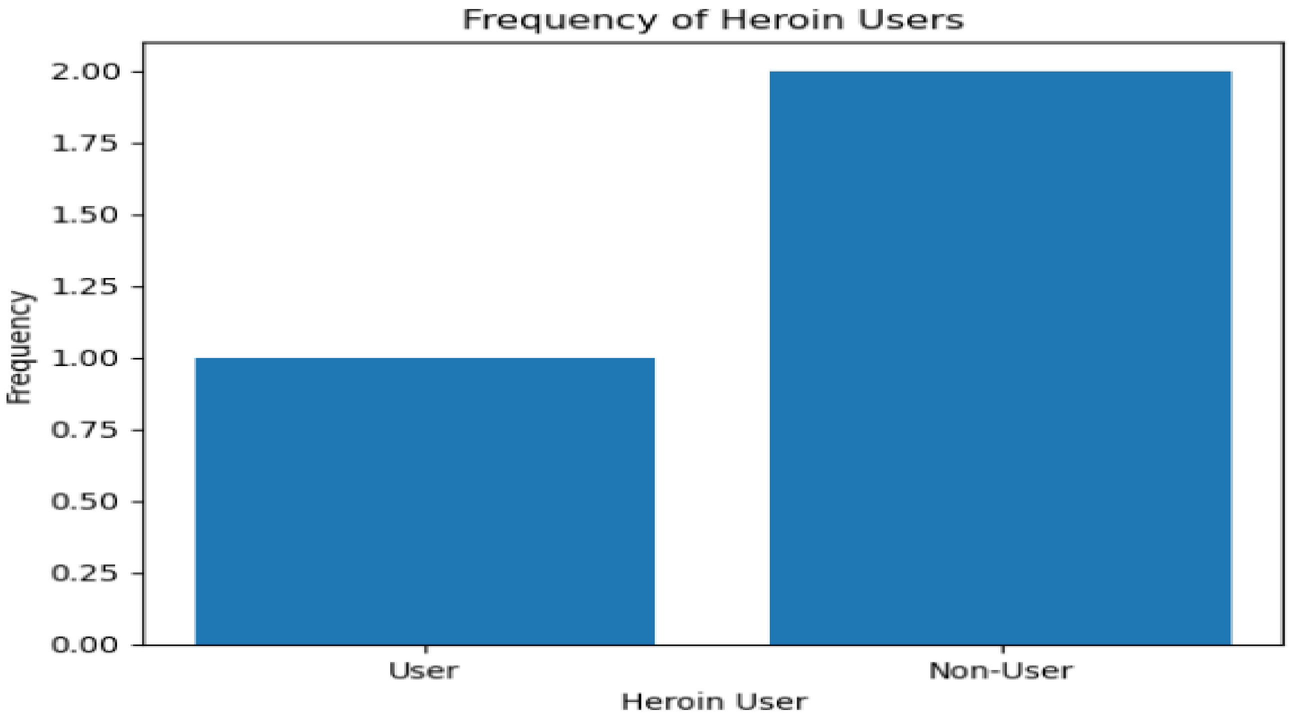
Model Evaluation (cocaine)



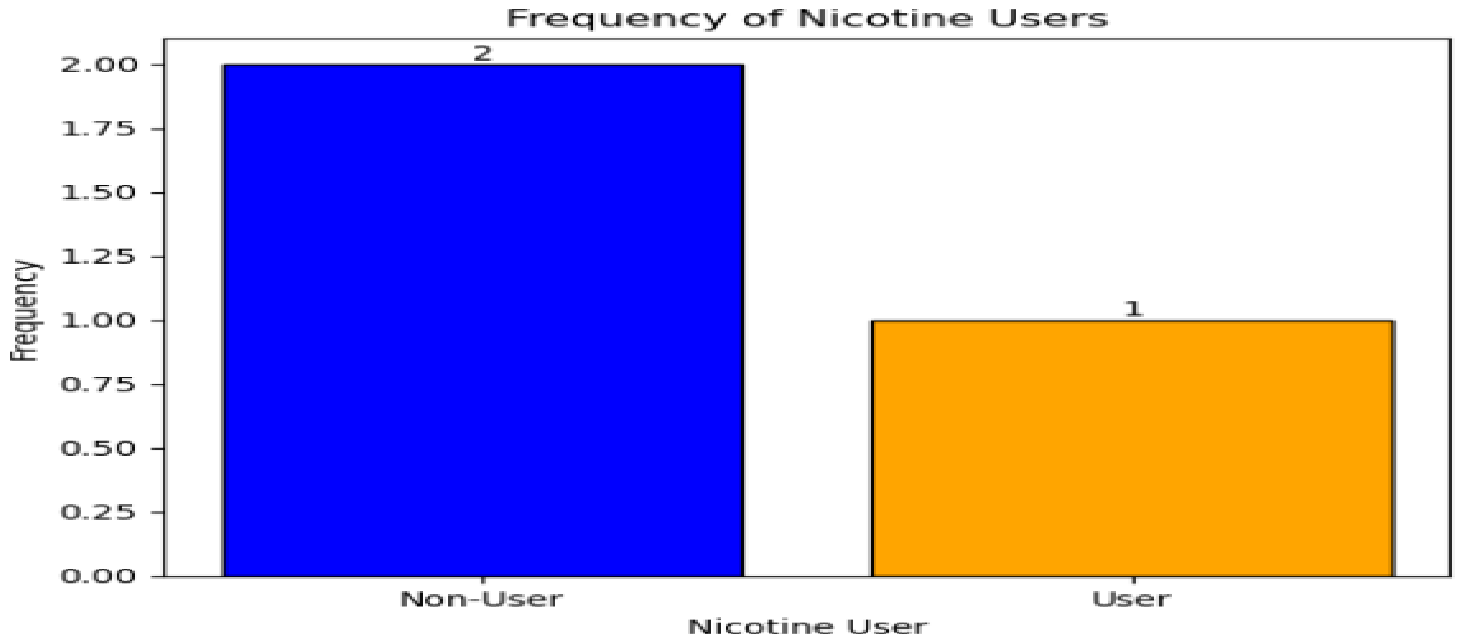
Model Evaluation (Methamphetamine)



Model Evaluation (Heroin)



Model Evaluation (Nicotine)



conclusion

Overall we see that our Logistic Regression, Random Forest Classifiers, and SVM's performed the best. The models performed best when classifying Cocaine and Nicotine. Although, this is probably due to the much larger sample size from these drugs compared to Heroin

Thank you