Project Description:

You are provided a dataset file, **overdrawn.csv**, which was collected from a survey to study sensation-seeking, risk-taking, and problematic financial behaviors of college students. There were close to 150 questions on the survey, but only four of these variables are included in this dataset.
The primary interest for the researchers was factors relating to whether or not a student had ever overdrawn a checking account.

Getting Ready

Before we start, you need a few things:

- **Python**: If you don't have Python, you can download it from **python.org**.
- **Jupyter Notebook (Optional)**: This tool is great for working with Python. You can install it with pip, which is a tool that helps you install Python stuff.

```
pip install jupyter
```

- Alternatively, you can also use Visual Studio Code IDE for code development.
- **Useful Library**: We'll use a special Python library called 'graphviz'. This library allows to draw a decsion tree in png format.

Installation

- This package runs under Python 3.8+, use pip to install:

```
pip install graphviz
```

- To render the generated DOT source code, you also need to install Graphviz (download page, archived versions, installation procedure for Windows). This can be done as:
    - You can also install it for Windows using the following links based on you have 32-bit or 64-bit machine:
        - Download the 32-bit 410 or 64-bit 7.3k exe file. When installation screen appears, select the option to set the Path variable by this installation.
            - Make sure that the directory containing the dot executable is on your systems' PATH (sometimes done by the installer; setting PATH on Linux, Mac, and Windows).
    - Documentation for graphviz: https://graphviz.readthedocs.io/en/stable/

Now that we have the tools ready, we can start creating our Classification Model.

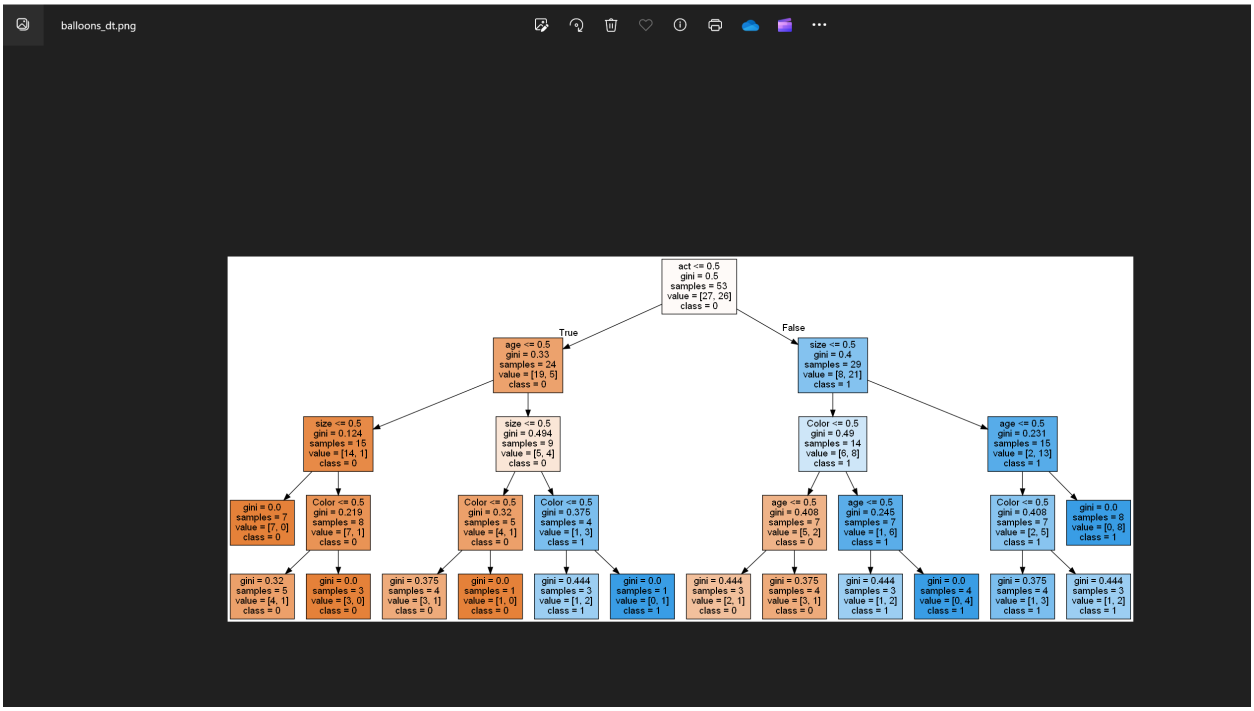
Step 1: Run the sample learning decision tree Python program

First try to run the provided sample Python program, **BalloonsDecisionTree.py** , which uses a dataset file, **balloons.csv**. Make sure both these files are located in the same folder.

Note: it is highly recommended to run this sample or your solution from the Windows command line.

It should look similar to the following:

```
>python BalloonsDecisionTree.py
0.6521739130434783
[[7 1]
 [7 8]]
```



Step 2: Creating Learning Decision Tree Classification Model

The provided dataset, **overdrawn.csv,** contains the following variables:

Age:	Age of the student in years
Sex:	0 = male or 1 = female
DaysDrink:	Number of days drinking alcohol (in past 30 days)
Overdrawn:	Has the student overdrawn a checking account? 0= no or 1 = yes

Your task is to create a decsion-tree based model to predict the student overdrawing from the checking account based on *Age, Sex, and DaysDrink.*

Note that since *DaysDrink* is a numeric variable, you may have to convert it into a categorical one. One suggestion for that would be (in Python):

```
# Change DaysDrink into categorical data
conditions = [
    (df['DaysDrink'] < 7),
    (df['DaysDrink'] >= 14),
    (df['DaysDrink'] >= 7) & (df['DaysDrink'] < 14)
]
categories = [0, 2, 1]

# Apply the conditions to create the categorical data
df['DaysDrink'] = np.select(conditions, categories)
```

Step 3: Developing the Python Code

Develop the Python code following the pattern provided in the sample Python program. Name your program file as ***StudentBehaviors.py***

Step 4: Run the program

Run your solution from the Windows command line as follows:

C:\workspace\ai\classification>**python StudentBehaviors.py**

Step 5: Predicting Values

Your code should be answering the below five predictions or queries using the decision tree classification program you developed:

1. Predict whether a 20-year-old male student who has drunk alcohol for 10 days in the past 30 days will overdraw a checking account.
2. Predict whether a 25-year-old female student who has drunk alcohol for 5 days in the past 30 days will overdraw a checking account.
3. Predict whether a 19-year-old male student who has drunk alcohol for 20 days in the past 30 days will overdraw a checking account.
4. Predict whether a 22-year-old female student who has drunk alcohol for 15 days in the past 30 days will overdraw a checking account.
5. Predict whether a 21-year-old male student who has drunk alcohol for 20 days in the past 30 days will overdraw a checking account.

The output of these queries should look like below:

```
C:\workspace\ai\classification>python StudentBehaviors.py
Accuracy: 0.9090909090909091
Confusion Matrix:
[[117   1]
 [ 11   3]]
Prediction 1: Will the student overdraw a checking account? No
Prediction 2: Will the student overdraw a checking account? No
Prediction 3: Will the student overdraw a checking account? No
Prediction 4: Will the student overdraw a checking account? Yes
Prediction 5: Will the student overdraw a checking account? No
```

In addition, your output should be able to display a decision tree for this problem (similar in nature but different in structure and values as that of the sample shown above

Step 6: Deliverables

You are required to submit a report containing the following deliverables:

1. **Introduction**:
    - Briefly introduce the problem statement and the dataset used.
    - Mention the objectives of the analysis, such as predicting whether a student will overdraw a checking account based on their age, gender, and alcohol consumption habits.
2. **Data Preprocessing**:
    - Describe any preprocessing steps performed on the dataset, such as handling missing values or converting categorical variables into numerical format.

- Explain the rationale behind each preprocessing step.

3. **Model Development**:
   - Explain the process of building the decision tree classification model.
   - Describe the features used in the model (age, gender, days of drinking).
4. **Model Evaluation**:
   - Present metrics such as accuracy score, and confusion matrix.
   - Interpret the results and discuss the effectiveness of the model in predicting whether a student will overdraw a checking account.
   - Print the decsion-tree at the output and take the screenshot and include it in the pdf file.
5. **Predictions and Queries**:
   - Provide Python code to implement five predictions or queries based on the trained model.
   - Explain each prediction/query and its relevance to the problem statement.
   - Display at the output screen the predictions obtained from the model for each query.
6. **Conclusion**:
   - Summarize the key findings of the analysis.
   - Discuss any limitations or assumptions made during the analysis.
7. **References**:
   - Include any references or resources used during the analysis.
8. **Submission Files:** submit only 2 files, one pdf and one zip file.
   - Ensure that your report is well-organized, clearly written, and includes appropriate visualizations and explanations and submit as a PDF file named like *yourname-project4-cis***.pdf*.
   - Provide Python code and dataset in a zip file named like *yourname-project4-cis***.zip*.