Homework 6

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd, numpy as np, matplotlib
%matplotlib inline
import sklearn
from sklearn import *
from sklearn.ensemble import *
import sklearn
%matplotlib inline
```

```
df = pd.read_csv("Data/covid19.csv", sep = ",")
df.shape, df.dtypes
df.head(3), df.tail(3)
df.isnull().sum()
df.describe(include = "all")
```

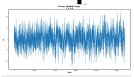
Problem 1

Problem 1 Data Analysis: The file Data/Titan.csv contains 1309 rows and 14 columns. Each row contains information about a passenger on the Titanic, including the passenger's name, sex, age, ticket number, passenger class, cabin, and whether the passenger survived (0 = no, 1 = yes). Your task is to explore the data, clean it, and prepare it for modeling. Specifically, you should:

1. Load the data into a DataFrame. Print the shape and dtypes of the dataset, and display the first and last rows. Identify which columns are numerical and which are categorical.
2. Check for missing values in each column. Report the number of missing values per column. Decide how you will handle missing values in Age, Cabin, and Embarked. Justify your choice in a few sentences.
3. Fill in missing values in Age with the median age of passengers in the same Pclass and Sex. Fill in missing Embarked with the most common value. Drop Cabin entirely.
4. Create a new column FamilySize = SibSp + Parch + 1. Create a binary column IsAlone that is 1 if FamilySize is 1, and 0 otherwise.
5. Create a new column Title by extracting the titles from the Name column (e.g., "Mr", "Mrs", "Miss", "Master"). Replace any rare titles (appearing fewer than 10 times) with "Rare".
6. Encode Sex and Title as numerical values using one-hot encoding. Drop the original Name, Ticket, and PassengerId columns.
7. Split the data into a training set (80%) and a test set (20%), stratified by Survived. Use random_state=42.
8. Scale the features using StandardScaler. Fit the scaler on the training set only, then transform both sets.
9. Train a logistic regression model on the training set. Report the accuracy on both the training set and the test set. Do you see any signs of overfitting?
10. Train a decision tree classifier (max_depth=5) on the same data. Report the accuracy on both the training set and the test set. Which model performs better on the test set? Which is more interpretable? Explain why.

```
df = pd.read_csv("Data/Titan.csv", sep = ",")
df.shape, df.dtypes
df.head(3), df.tail(3)
df.isnull().sum()
df.describe(include = "all")
```

```
```




```
# ---------------------------------------------------------------
# 1. Generate random frames (2000 frames, 16x16)
# ---------------------------------------------------------------
np.random.seed(42)
num_frames = 2000
frames = np.random.randint(0, 16, size=(num_frames, 16, 16))

# ---------------------------------------------------------------
# 2. Create figure
# ---------------------------------------------------------------
fig = plt.figure(
    figure_ref="1",
    figure_type="chart",
    figure_usefulness="7",
    figure_caption="Frame 2000 of 2000",
    figure_box="0.464,0.000,0.518,0.041",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.039,0.545,0.073",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.144,0.518,0.185",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.183,0.545,0.217",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.288,0.518,0.329",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.327,0.545,0.361",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.432,0.518,0.473",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.471,0.545,0.505",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.576,0.518,0.617",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.615,0.545,0.649",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.720,0.518,0.761",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.759,0.545,0.793",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.864,0.518,0.905",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.903,0.545,0.937",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,0.992,0.518,1.000",
    figure_text="Frame 2000 of 2000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,1.000,0.545,1.000",
    figure_text="0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Frame",
    figure_rotation="0",
    figure_kind="chart",
    figure_usefulness="7",
    figure_box="0.464,1.000
```