METHODOLOGY

The objective was to evaluate the randomness of binary sequences using NIST-defined statistical tests, and to explore the use of AI to classify sequences as random or non-random based on test outputs.

The following methodology was adopted:

1. **Binary Sequence Generation**
2. **NIST Test Suite Execution**
3. **Result Extraction**
4. **Creating Dataset**
5. **Dataset Refinement**
6. **Initial AI Model Setup**

BINARY SEQUENCE GENERATION

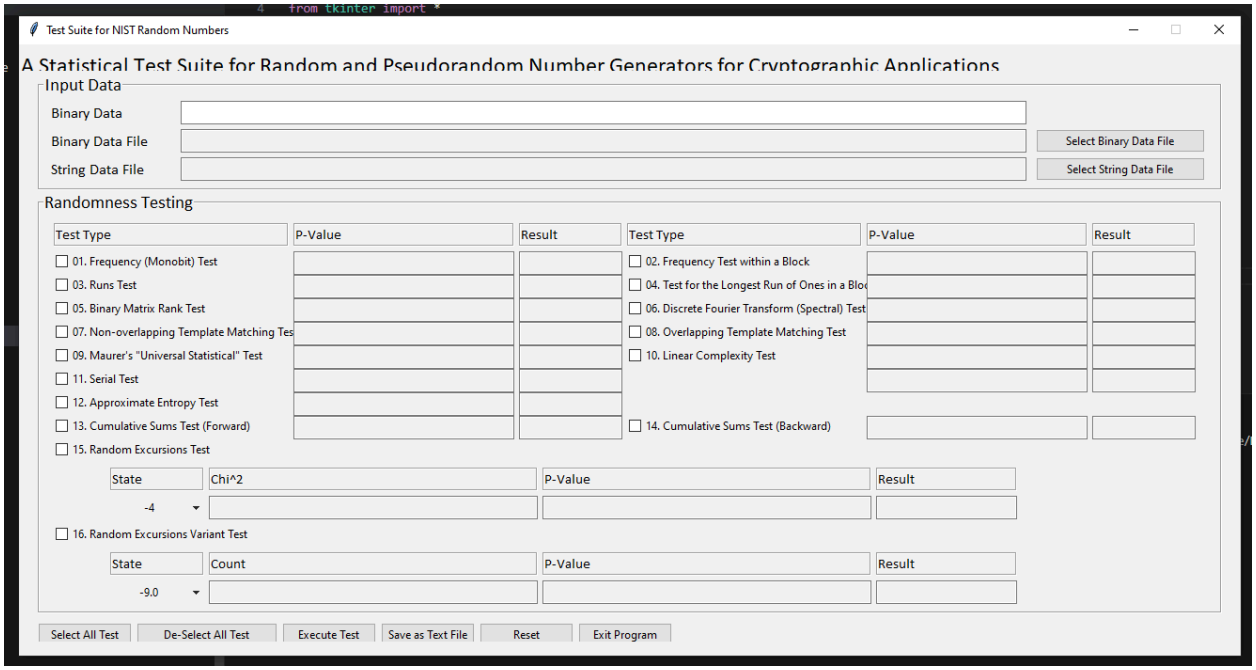
A total of 40 binary sequences were generated — 10 random and 10 non-random.

- Random sequences were created using Python's `random` module.
- Non-random sequences were constructed using repetitive or biased patterns.

NIST TEST SUITE EXECUTION

Each binary sequence was tested using the **NIST Statistical Test Suite**.

- The GUI version of the NIST test suite was used for simplicity.
- Each sequence was input individually, and the results were saved.



- Input Data - Input Data contains Binary Data, Binary Data File and String Data File
- Binary Data - You can only enter a BINARY STRING here (ex:11001001000011111101101010100010001000010110100011000010001101001100010011000110011000101000101110000)
- You can select the type of test you want to perform by clicking the corresponding checkbox or press "Select All Test" to select everything
- You can cancel the selection by clicking the corresponding checkbox or press "De-Select All Test" to cancel everything
- Once you have your data ready and selected the test you want to perform, then you can press "Execute Test" button to execute the test

What is P-VALUE

P-value tells us how likely it is that the sequence is random.
It's a number between 0 and 1.
> 0.01 Pass: sequence looks random
< 0.01 Fail: sequence is probably not random

WHAT ARE THESE TESTS ?

1. **Frequency (Monobit) Test**
   Evaluates whether the number of ones and zeros in the sequence are approximately equal, as expected in a random sequence.

2. **Frequency Test within a Block**
   Divides the sequence into blocks and checks the uniformity of ones and zeros within each block. Imbalances in smaller sections may indicate local patterns.

3. **Runs Test**
   Examines the total number of uninterrupted sequences (runs) of identical bits. Deviations from expected run lengths suggest predictability.

4. **Test for the Longest Run of Ones in a Block**
   Measures the length of the longest sequence of consecutive ones in fixed-size blocks. Unusually long or short runs are indicative of non-randomness.

5. **Binary Matrix Rank Test**
   Converts the sequence into binary matrices and computes their ranks. A deficiency in full-rank matrices implies redundancy and non-random structure.

6. **Discrete Fourier Transform (Spectral) Test**
   Analyzes the frequency components of the sequence to detect periodic patterns that suggest deviation from randomness.

7. **Non-overlapping Template Matching Test**
   Searches for a specified pattern in the sequence without allowing overlaps. Significant deviations in pattern occurrences suggest structural repetition.

8. **Overlapping Template Matching Test**
   Similar to the non-overlapping version but permits overlapping occurrences of patterns, making it more sensitive to repeated sub-patterns.

9. **Maurer’s Universal Statistical Test**
   Determines the compressibility of the sequence. Random data is generally incompressible, so high compressibility indicates non-randomness.

10. **Linear Complexity Test**
    Assesses the complexity of the sequence by determining the length of the shortest linear feedback shift register that can generate it. Low complexity suggests predictability.

11. **Serial Test**
    Measures the frequency of all possible overlapping m-bit patterns within the sequence to identify any imbalance or repetitive behavior.

12. **Approximate Entropy Test**
    Quantifies the unpredictability of the sequence by evaluating the frequency of overlapping patterns. Lower entropy implies a lack of randomness.

13. **Cumulative Sums Test (Forward and Backward)**
    Performs a random walk based on the binary sequence, interpreting 1 as +1 and 0 as -1. It observes the cumulative sum and checks for large deviations from zero, which would indicate bias.

14. **Random Excursions Test**
    Tracks the number of visits to specific states (such as +1, -1, +2, etc.) in a cumulative sum random walk. Anomalies in visit frequency suggest structural patterns.

15. **Random Excursions Variant Test**
    Similar to the Random Excursions Test but considers the total number of visits to each state without restricting to cycles. It provides a more comprehensive view of the behavior of the sequence.

RESULT EXTRACTION

Manual Execution and Observation

Each sequence was individually tested using the NIST GUI tool. Specific tests (e.g., Frequency Test, Runs Test, Approximate Entropy Test) were selected based on their relevance and applicability.

For each test executed, the **P-value** and the **pass/fail outcome** were recorded.

P > 0.01 → Test Passed (likely random)

P ≤ 0.01 → Test Failed (likely not random)

Type of Test	P-Value	Conclusion
01. Frequency (Monobit) Test	0.0	Non-Random
02. Frequency Test within a Block	0.0	Non-Random
03. Runs Test	0.0	Non-Random
04. Test for the Longest Run of Ones in a Block	2.7642484974045884e-22	Non-Random
05. Binary Matrix Rank Test	2.138219156207687e-13	Non-Random
06. Discrete Fourier Transform (Spectral) Test	9.961249943714423e-100	Non-Random
07. Non-overlapping Template Matching Test	0.915418422483985	Random
08. Overlapping Template Matching Test	0.2920734664158972	Random
09. Maurer's "Universal Statistical" Test	-1.0	Non-Random
10. Linear Complexity Test	3.270538383604854e-97	Non-Random
11. Serial Test:		
	0.0	Non-Random
	0.0	Non-Random
12. Approximate Entropy Test	0.0	Non-Random
13. Cumulative Sums Test (Forward)	0.0	Non-Random
14. Cumulative Sums Test (Backward)	0.0	Non-Random
15. Random Excursions Test:		

State	Chi Squared	P-Value	Conclusion
-4	28.851728446480635	2.4793357264148452e-05	Non-Random
-3	14.462857142857143	0.0129217965533314	Random
-2	14.587301587301589	0.01227925191961656	Random
-1	1.5714285714285714	0.9046827003530795	Random
+1	6.428571428571429	0.2667207596069264	Random
+2	6.319223985890653	0.27638433726342715	Random
+3	7.9952000000000005	0.15650036346871068	Random
+4	4.880287975248408	0.4306641688300008	Random

To prepare the data for use in machine learning models, the qualitative outcomes were converted into binary form:

- Test Passed → `1`
- Test Failed → `0`

```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,0
1,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0
1,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0
0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
1,1,0,0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,1,1
```

Inconsistent test outputs (e.g., missing P-values)

Removal of any corrupted or partial rows from the dataset

```
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
```

AI MODEL SETUP

- We planned to use **supervised machine learning**, where the model learns from labeled examples (random or not random) and predicts labels for new sequences.
- Model Selection
  We are using a **Logistic Regression model**, a commonly used **binary classification** algorithm.
- Definition
  A statistical model that uses a logistic function to estimate the probability of a binary outcome (e.g., random or non-random). Despite its name, it is used for classification tasks, not regression.

- **Other possible Models**

  1. Decision Trees
  A non-parametric model that splits data into branches based on feature values, forming a tree-like structure. Each leaf node represents a class label.

  2. Random Forests
  An ensemble model that builds multiple decision trees and combines their outputs to improve generalization and reduce overfitting.

  4. Neural Networks
  Models inspired by the human brain, consisting of layers of interconnected nodes that learn complex patterns from data.

TRAINING AND TESTING

- The dataset is split into a **training set** and a **testing set**.

- The model learns patterns from the training set (i.e., how P-values relate to randomness).

- The model's performance is evaluated on the testing set using accuracy, precision, and recall scores.