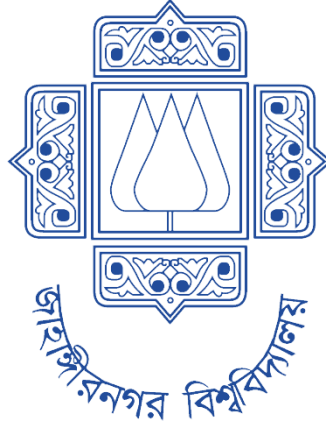


**Institute of Information Technology (IIT)
Jahangirnagar University**



Course Code: MICT 5101
Course Title: Probability and Stochastic Process

Assignment - 01

Submitted to:
Dr. Md. Rezaul Karim
Professor
Department of Statistics and Data Science
Jahangirnagar University

Submitted by:
Name: Md. Shakil Hossain
Roll No: 1061 (2023)
5th Year 1st Semester
IIT, JU

Submission Date: 19/12/2024

Name:

Predicting Student Grade Progression through Markov Chain Analysis and Stationary Distribution.

Aim:

The purpose of this experiment is to use the concepts of Markov Chains to examine the dynamics of student academic achievement. The study aims to simulate the probability of grade transitions between semesters by building a transition probability matrix using historical grade data. Understanding the long-term behavior of student grades through an analysis of the stationary distribution of this Markov Chain may help identify patterns and trends that might guide academic interventions and enhance student achievement.

Software:

- Google Sheets
- Google Colab

Theory:

Stochastic Process:

A stochastic process is said to be the Markov chain if the conditional probability distribution of future states depends only on the present state, not on past states. A discrete-time stochastic process $\{X_n\}$ $n \geq 0$ is said to be a discrete Markov chain if the conditional distribution of any future state X_{n+1} , given the past states X_0, X_1, \dots, X_{n-1} and the present state X_n , is independent of the past states and depends only on the current state.

That is

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0) = P(X_{n+1} = j \mid X_n = i)$$

for all $n \geq 0$ and all $i_0, i_1, \dots, i_{n-1}, i, j \in S$.

Transition Probability:

A transition probability refers to the probability of moving from one state to another in a Markov chain. It quantifies how likely a system will transition from a current state i to a next state j in a given time step. The transition probability from state i to state j is denoted as:

$$P_{ij} = P(X_{n+1} = j \mid X_n = i)$$

Properties:

1. Non-negativity: $P_{ij} \geq 0$ for all states i and j .
2. Normalization: $\sum_{j \in S} P_{ij} = 1$

This ensures that the system will transition to some state j with certainty from any state i .

Methodology:

This experiment uses a quantitative approach to investigate how student academic performance is influenced by Markov Chain theory. The following phases are included in it:

Data Acquisition:

Historical grade data covering a minimum of eight semesters is collected from a cohort of students. The data set is painstakingly vetted to encompass a comprehensive spectrum of grades.

Data Preprocessing:

The collected data undergoes meticulous organization and transformation. Sequential grade changes for each student across consecutive semesters are calculated. Subsequently, a discrete state space is established by assigning a unique numerical identifier to each distinct grade.

Transition Matrix Construction:

A transition probability matrix (P) is meticulously constructed. Each element within this matrix, denoted as P_{ij} , represents the probability of transitioning from grade ' i ' to grade ' j '. The calculation of P_{ij} is achieved by dividing the frequency of transitions from grade ' i ' to grade ' j ' by the total occurrences of grade ' i '.

Markov Chain Properties Assessment:

A rigorous assessment of the constructed Markov Chain is conducted to ascertain its irreducibility and aperiodicity, thereby ensuring the validity of long-term predictions.

Stationary Distribution Computation:

The stationary distribution (π) of the Markov Chain is meticulously computed by solving the equation $\pi P = \pi$, where π represents a probability vector embodying the long-term equilibrium distribution of grades. Numerical methods or specialized computational tools are employed to effectively solve this system of equations.

Data Analysis and Interpretation:

A comprehensive analysis of the computed stationary distribution is undertaken to identify discernible long-term trends and patterns within the distribution of student grades. The findings are meticulously interpreted within the broader context of student academic performance, with a particular focus on their potential implications for the implementation of targeted educational interventions.

3. Construct the Transition Probability Matrix

Code:

```
for _, row in transition_counts.iterrows():
    i, j = state_space[row['From']], state_space[row['To']]
    P[i, j] = row['Count']

P = P / P.sum(axis=1, keepdims=True)
print("Transition Probability Matrix (P):")
print(pd.DataFrame(P, index=grades, columns=grades))
```

Output:

Transition Probability Matrix (P):						
	A	A+	A-	B	B+	B-
A	0.656716	0.074627	0.164179	0.000000	0.104478	0.000000
A+	0.666667	0.000000	0.166667	0.000000	0.166667	0.000000
A-	0.244755	0.013986	0.496503	0.027972	0.216783	0.000000
B	0.017544	0.035088	0.140351	0.298246	0.473684	0.035088
B+	0.064748	0.000000	0.410072	0.136691	0.366906	0.021583
B-	0.000000	0.000000	0.000000	0.625000	0.250000	0.125000

4. Check the Nature of the State

Code:

```
def check_irreducibility(matrix):
    n_components, _ = connected_components(csgraph=matrix > 0,
    directed=True, connection='strong')
    return n_components == 1

def check_aperiodicity(matrix):
    periods = []
    for i in range(len(matrix)):
        reachable = np.nonzero(matrix[i, :] > 0)[0]
        if len(reachable) > 0:
            steps = []
            for j in reachable:
                steps.append(gcd(i + 1, j + 1))
            periods.append(gcd(*steps))
    return all(p == 1 for p in periods)

irreducible = check_irreducibility(P)
print("\nIrreducible:", irreducible)
aperiodic = check_aperiodicity(P)
print("Aperiodic:", aperiodic)
```

Output:

```
Irreducible: True
Aperiodic: True
```

5. Compute the Stationary Distribution

Code:

```
I = np.identity(len(P))
A = P.T - I
A_augmented = np.vstack([A, np.ones(len(P))])
b_augmented = np.zeros(len(P))
b_augmented = np.append(b_augmented, 1)
pi = np.linalg.lstsq(A_augmented, b_augmented, rcond=None)[0]
print("Steady-state probabilities (pi):", pi)
```

Output:

```
Steady-state probabilities (pi): [0.34118302 0.03228616 0.32564301 0.06470334 0.22796682 0.00821765]
```

Interpretation and Analysis:

After completing the stationary distribution

The End