**Institute of Information Technology (IIT)**

**Jahangirnagar University**



**Course Code:** MICT 5101

**Course Title:** Probability & Stochastic Process

Assignment - 01

**Submitted to:**

Professor Dr. Md. Rezaul Karim

Department of Statistics and Data Science

**Submitted by:**

Name: Afrin Faria

Roll No: 1053 (MSc), 1981(BSc)

5th Year 1st Semester- MSc

**Submission Date:** 20/12/2024

## 

## **Name:**

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

## **Aim:**

This experiment aims to investigate the dynamics of student academic performance within the framework of Markov Chain theory. By constructing a transition probability matrix based on historical grade data, the analysis seeks to model the likelihood of transitions between different academic achievement levels. The analysis focuses on identifying long-term trends in student performance by determining the stationary distribution of the Markov Chain. The findings of this assignment are expected to provide valuable insights that can inform the development of targeted academic interventions and contribute to the improvement of student success.

## **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 {Xn} n≥0 is said to be a discrete Markov chain if the conditional distribution of any future state Xn+1, gives the past states X0,X1, . . . ,Xn−1 and the present state Xn, is independent of the past states and depends only on the current state.

That is

## P(Xn+1 = j ∣ Xn = i,Xn−1 = in−1, . . . ,X1 = i1,X0 = i0) = P(Xn+1 = j ∣ Xn = i)

## for all n ≥ 0 and all i0, i1, . . . , in−1, i, j ∈ 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:

Pij = P(Xn+1 = j S Xn = i)

**Properties:**

1. Non-negativity: Pij ≥ 0 for all states i and j.

2. Normalization:

j∈S Pij = 1

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

## **Methodology:**

This experiment utilizes a quantitative approach to investigate the dynamics of student academic performance within the framework of Markov Chain theory. The methodology encompasses the following stages:

1. **Data Collection:** Historical grade data spanning a minimum of eight semesters is meticulously collected from a cohort of students, ensuring a comprehensive representation of the grade spectrum.
2. **Data Preprocessing:** The collected data undergoes rigorous organization and transformation. Sequential grade changes for each student across consecutive semesters are calculated, and subsequently, a discrete state space is established by assigning a unique numerical identifier to each distinct grade level.
3. **Transition Matrix Construction:** A transition probability matrix (P) is meticulously constructed. Each element within this matrix, denoted as Pij, represents the probability of transitioning from grade 'i' to grade 'j'. The calculation of Pij is achieved by dividing the frequency of transitions from grade 'i' to grade 'j' by the total occurrences of grade 'i'.
4. **Markov Chain Properties Assessment:** The constructed Markov Chain undergoes a rigorous assessment to ascertain its irreducibility and aperiodicity, thereby ensuring the validity of long-term predictions.
5. **Stationary Distribution Computation:** The stationary distribution (π) of the Markov Chain is meticulously computed by solving the equation πP = π, 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.
6. **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.

## **1. Necessary Libraries**

## **Code:**

| import numpy as np  import pandas as pd  from scipy.sparse.csgraph import connected\_components  from math import gcd  from numpy.linalg import eig |
| --- |

## **2. Pre-Processing**

## **Code:**

| import pandas as pd  def extract\_grade\_transitions\_v2(file\_path):    grade\_data = pd.read\_excel(file\_path)  grade\_columns = grade\_data.columns[1:] # Exclude the first column (e.g., student ID)  grade\_data[grade\_columns] = grade\_data[grade\_columns].astype(str)  all\_transitions = []  for \_, student\_grades in grade\_data.iterrows():  for i in range(len(grade\_columns) - 1):  prev\_grade = student\_grades[grade\_columns[i]]  next\_grade = student\_grades[grade\_columns[i + 1]]  if prev\_grade != 'F' and next\_grade != 'F':  all\_transitions.append((prev\_grade, next\_grade))  transition\_counts = pd.DataFrame(all\_transitions, columns=['Previous\_Grade', 'Next\_Grade']).value\_counts().reset\_index()  transition\_counts.columns = ['Previous\_Grade', 'Next\_Grade', 'Count']  return transition\_counts  file\_path = 'prob.xlsx'  grade\_transitions = extract\_grade\_transitions\_v2(file\_path)  print(grade\_transitions) |
| --- |

## **Output:**

|  |
| --- |

## **3. Construct the Transition Probability Matrix**

## **Code:**

| def create\_transition\_matrix(transition\_counts, state\_space):  num\_states = len(state\_space)  transition\_matrix = np.zeros((num\_states, num\_states))    state\_index\_map = {state: i for i, state in enumerate(state\_space)}    for \_, row in transition\_counts.iterrows():  from\_state = row['From']  to\_state = row['To']  from\_index = state\_index\_map[from\_state]  to\_index = state\_index\_map[to\_state]  transition\_matrix[from\_index, to\_index] = row['Count']    row\_sums = transition\_matrix.sum(axis=1)  row\_sums[row\_sums == 0] =  transition\_matrix = transition\_matrix / row\_sums[:, np.newaxis]  return transition\_matrix  state\_space = ['A+', 'A', 'A-', 'B+', 'B', 'B-']  transition\_matrix = create\_transition\_matrix(transition\_counts, state\_space)  print("Transition Probability Matrix (P):")  print(pd.DataFrame(transition\_matrix, index=state\_space, columns=state\_space)) |
| --- |

## **Output:**

|  |
| --- |

## **4. Check the Nature of the State**

## **Code:**

| import numpy as np  from scipy.sparse.csgraph import connected\_components  from math import gcd  def is\_irreducible(transition\_prob\_matrix):  num\_components, \_ = connected\_components(csgraph=(transition\_prob\_matrix > 0), directed=True, connection='strong')  return num\_components == 1  def is\_aperiodic(transition\_prob\_matrix):  num\_states = transition\_prob\_matrix.shape[0]  periods = []  for current\_state in range(num\_states):  reachable\_states = np.where(transition\_prob\_matrix[current\_state, :] > 0)[0]  if len(reachable\_states) > 0:  steps = [gcd(current\_state + 1, next\_state + 1) for next\_state in reachable\_states]  periods.append(gcd(\*steps))  return all(period == 1 for period in periods)  is\_irreducible\_chain = is\_irreducible(transition\_prob\_matrix)  print("\nIrreducible:", is\_irreducible\_chain)  is\_aperiodic\_chain = is\_aperiodic(transition\_prob\_matrix)  print("Aperiodic:", is\_aperiodic\_chain) |
| --- |

## 

## 

## **Output:**

|  |
| --- |

## **5. Compute the Stationary Distribution**

## **Code:**

| def calculate\_stationary\_distribution(transition\_matrix):    num\_states = transition\_matrix.shape[0]  identity\_matrix = np.identity(num\_states)    A = transition\_matrix.T - identity\_matrix  ones\_vector = np.ones(num\_states)  augmented\_matrix = np.vstack([A, ones\_vector])    b\_vector = np.zeros(num\_states)  augmented\_vector = np.append(b\_vector, 1)  stationary\_distribution = np.linalg.lstsq(augmented\_matrix, augmented\_vector, rcond=None)[0]  return stationary\_distribution  stationary\_dist = calculate\_stationary\_distribution(transition\_matrix)  print("Steady-state probabilities (pi):", stationary\_dist) |
| --- |

## 

## **Output:**

|  |
| --- |

## **Interpretation and Analysis:**

**1. Irreducibility and Aperiodicity:**

* Irreducible: Since the Markov chain is irreducible, it implies that it is possible for a student to transition from any grade to any other grade within a finite number of semesters. This suggests that the system is not trapped in any particular subset of grades and that students can potentially improve or decline in their academic performance over time.
* Aperiodic: The aperiodicity property indicates that there is no fixed pattern or cycle in the grade transitions. This means that students are not likely to oscillate between the same grades repeatedly.

**2. Steady-State Probabilities:**

* The steady-state probabilities (π) represent the long-term equilibrium distribution of student grades. In other words, if the system were to evolve for a very long time, the proportion of students in each grade would converge to these probabilities.
* Interpretation of Results:  
  + Highest Probability: The grade with the highest steady-state probability (e.g., 'B' in this example) suggests that, in the long run, the largest proportion of students are likely to be in that grade.
  + Grade Distribution: The distribution of probabilities across the grades provides insights into the overall academic performance of the student population. A more concentrated distribution around higher grades would indicate a generally high-achieving student body.
  + Implications: The steady-state probabilities can be used to:
    - Identify areas of concern: Focus on providing additional support to students in grades with lower long-term probabilities.
    - Inform academic policies: Adjust academic support programs and resources based on the predicted long-term grade distribution.
    - Set realistic academic expectations: Establish benchmarks and goals based on the expected long-term performance of students.

**3. Limitations and Considerations:**

* Data Limitations: The accuracy of the analysis depends heavily on the quality and quantity of the data used.
* External Factors: The model may not fully capture the influence of external factors such as student motivation, personal circumstances, and external support systems, which can significantly impact academic performance.
* Dynamic Nature: The academic environment is dynamic and can change over time. The model may need to be updated periodically to reflect changes in curriculum, teaching methods, and student demographics.

Overall, the Markov Chain analysis provides valuable insights into the long-term behavior of student grades. By understanding the transition probabilities and the steady-state distribution, one can make more informed decisions regarding academic support, resource allocation, and the development of effective interventions to improve student success.