



# Analytics e Inteligência Artificial: Implementações Práticas e Técnicas com Java

A Inteligência Artificial (IA) e o Analytics desempenham um papel essencial na transformação digital em diversos setores, como saúde, educação, varejo e indústria.

Este documento aborda as principais aplicações práticas desses conceitos, explorando temas como IA generativa e sistemas de recomendação, com exemplos totalmente funcionais e implementados em **Java**.

O objetivo é fornecer uma visão técnica detalhada, com **inputs interativos** e **outputs esperados**, permitindo que desenvolvedores, pesquisadores e profissionais de TI compreendam e implementem soluções reais.

## 1. IA Generativa: Modelo de Bigramas

IA generativa cria novos conteúdos, como textos, imagens ou músicas, aprendendo padrões a partir de um conjunto de dados.

Neste exemplo, implementamos um modelo de bigramas, que prevê a próxima palavra com base na anterior. O modelo é construído a partir de um texto de treinamento fornecido pelo usuário.

```
import java.util.HashMap;
import java.util.Map;
import java.util.Random;
import java.util.Scanner;

public class TextGenerator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Entrada do usuário
        System.out.println("Digite o texto de treinamento:");
        String trainingText = scanner.nextLine();
        System.out.println("Digite a palavra inicial:");
        String startWord = scanner.nextLine();
        System.out.println("Digite o número de palavras para gerar:");
        int wordCount = scanner.nextInt();

        // Construção do modelo de bigramas
        Map<String, String[]> bigrams = buildBigrams(trainingText);
```



```
// Geração de texto
String generatedText = generateText(bigrams, startWord, wordCount);
System.out.println("Texto gerado: " + generatedText);
}

private static Map<String, String[]> buildBigrams(String text) {
    Map<String, String[]> bigrams = new HashMap<>();
    String[] words = text.split(" ");
    for (int i = 0; i < words.length - 1; i++) {
        bigrams.putIfAbsent(words[i], new String[]{});
        bigrams.put(words[i], append(bigrams.get(words[i]), words[i + 1]));
    }
    return bigrams;
}

private static String generateText(Map<String, String[]> bigrams, String startWord, int length)
{
    StringBuilder text = new StringBuilder(startWord);
    Random random = new Random();
    String currentWord = startWord;

    for (int i = 1; i < length; i++) {
        String[] nextWords = bigrams.getOrDefault(currentWord, new String[]{});
        if (nextWords.length == 0) break;
        currentWord = nextWords[random.nextInt(nextWords.length)];
        text.append(" ").append(currentWord);
    }
    return text.toString();
}

private static String[] append(String[] array, String word) {
    String[] newArray = new String[array.length + 1];
    System.arraycopy(array, 0, newArray, 0, array.length);
    newArray[array.length] = word;
    return newArray;
}
}
```

## Exemplo de Execução

- **Input do usuário:**

less

Digite o texto de treinamento: A IA está transformando o mundo. IA gera texto, imagens e muito mais.

Digite a palavra inicial: A

Digite o número de palavras para gerar: 10

- **Output esperado:**

less

Texto gerado: A IA está transformando o mundo. IA gera texto imagens



## 2. Sistema de Recomendação com Similaridade de Cosseno

Sistemas de recomendação personalizam experiências de usuários com base em preferências. Aqui, implementamos um sistema que calcula a similaridade entre usuários, com base em suas avaliações de itens, utilizando a fórmula de similaridade de cosseno.

```
import java.util.Arrays;
import java.util.Scanner;

public class RecommendationSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Entrada: Matriz de avaliações
        System.out.println("Digite o número de usuários:");
        int numUsers = scanner.nextInt();
        System.out.println("Digite o número de itens:");
        int numItems = scanner.nextInt();
        int[][] ratings = new int[numUsers][numItems];

        System.out.println("Insira as avaliações dos usuários (0 ou 1):");
        for (int i = 0; i < numUsers; i++) {
            System.out.println("Usuário " + i + ":");
            for (int j = 0; j < numItems; j++) {
                ratings[i][j] = scanner.nextInt();
            }
        }

        System.out.println("Digite o índice do usuário-alvo:");
        int userIndex = scanner.nextInt();

        // Cálculo de similaridades
        double[] similarities = calculateSimilarities(ratings, userIndex);
        System.out.println("Similaridades com outros usuários: " + Arrays.toString(similarities));
    }

    private static double[] calculateSimilarities(int[][] ratings, int userIndex) {
        double[] similarities = new double[ratings.length];
        for (int i = 0; i < ratings.length; i++) {
            if (i != userIndex) {
                similarities[i] = cosineSimilarity(ratings[userIndex], ratings[i]);
            }
        }
        return similarities;
    }

    private static double cosineSimilarity(int[] user1, int[] user2) {
        double dotProduct = 0.0, normUser1 = 0.0, normUser2 = 0.0;
        for (int i = 0; i < user1.length; i++) {
            dotProduct += user1[i] * user2[i];
            normUser1 += Math.pow(user1[i], 2);
            normUser2 += Math.pow(user2[i], 2);
        }
    }
}
```



```
}  
    return dotProduct / (Math.sqrt(normUser1) * Math.sqrt(normUser2));  
}  
}
```

## Exemplo de Execução

- **Input do usuário:**

```
yaml  
Digite o número de usuários: 3  
Digite o número de itens: 4  
Insira as avaliações dos usuários:  
Usuário 0: 1 1 0 0  
Usuário 1: 1 0 0 1  
Usuário 2: 0 1 1 1  
Digite o índice do usuário-alvo: 0
```

- **Output esperado:**

```
less  
Similaridades com outros usuários: [0.0, 0.5, 0.4082482904638631]
```

## 3. Ambiente e Configuração

### Dependências

1. **Java Development Kit (JDK):** Recomendado JDK 11 ou superior.
2. **IDE (opcional):** IntelliJ IDEA, Eclipse ou qualquer editor de texto.

### Execução

1. Salve os códigos acima em arquivos .java.
2. Compile o arquivo com:

```
bash  
javac NomeDoArquivo.java
```

3. Execute com:

```
bash  
java NomeDoArquivo
```

### Links úteis

- <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>
- <https://docs.oracle.com/en/java/>



Este documento apresentou exemplos práticos e totalmente funcionais de **IA generativa** e **sistemas de recomendação** usando **Java**.

Ambos os exemplos oferecem uma base sólida para implementar soluções de inteligência artificial em cenários reais.

Com inputs dinâmicos e outputs verificáveis, os códigos estão prontos para uso e adaptação, atendendo a diferentes necessidades e níveis de complexidade.

Essas implementações permitem explorar o potencial do **Analytics** e da **IA**, promovendo a inovação e o desenvolvimento de soluções tecnológicas avançadas.

***EducaCiência FastCode para a comunidade***