UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

Sistema para recomendação de grade de disciplinas utilizando Fatoração Matricial Não Negativa

Igor Augusto Passos

Orientador
Geiza Maria Hamazaki da Silva

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2021

Catalogação informatizada pelo autor

P289

Passos, Igor Augusto
Sistema para recomendação de grade de disciplinas utilizando Fatoração Matricial Não Negativa / Igor Augusto Passos. -- Rio de Janeiro, 2021.
54

Orientadora: Geiza Maria Hamazaki da Silva.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2021.

1. Sistemas de recomendação. 2. Aprendizado de máquina. 3. Fatoração matricial não negativa. I. Silva, Geiza Maria Hamazaki da, orient. II. Título.

Sistema para Recomendação de grade de disciplinas utilizando Fatoração Matricial Não Negativa

Igor Augusto Passos

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado por:

_____

Profª Geiza Maria Hamazaki da Silva, D. Sc (UNIRIO)

_____

Prof. Pedro Nuno de Souza Moura, D. Sc (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

SETEMBRO DE 2021

Agradecimentos

Agradecer é uma tarefa muito difícil. Não por não querer agradecer, mas são tantas pessoas maravilhosas na minha vida que fica difícil agradecer o suficiente a todos. Poderia escrever páginas e páginas de agradecimentos que provavelmente ainda faltaria alguém.

Primeiramente aos meus pais Celso e Elisabete e minha irmã Aline pelo apoio sempre incondicional, principalmente em relação aos estudos. Poderia escrever milhares de páginas de agradecimento mas acho que nunca seria suficiente. Amo vocês.

Aos meus amigos Samantha, Rodrigo, Renata, Thaís e Yuri pelo carinho e por ouvirem tanto meus lamentos em relação à graduação, e por me ajudarem a segurar a barra em tantas outras situações da vida.

Às minhas madrinhas, por toda força e apoio nos momentos em que eu quase desisti.

A todos os meus amigos da faculdade, que tornaram minha permanência na UNIRIO um pouco mais feliz.

À Geiza, pela paciência e compreensão na orientação desse trabalho.

Por último, mas não menos importante, à UNIRIO e aos outros professores que de alguma forma passaram pelo meu caminho nessa formação. Entrar numa universidade federal em uma cidade grande me abriu várias portas que nunca pensei serem possíveis.

RESUMO

A oferta crescente de conteúdo nos últimos anos fez com que a escolha por itens relevantes tenha se tornado uma tarefa cada vez mais complexa, motivando o surgimento dos sistemas de recomendação. A popularização do *e-commerce* e os serviços de *streaming* fez com que esse tipo de sistema esteja mais presente no dia a dia do que nunca. Sua aplicação na sugestão de filmes, música, notícias e até mesmo em materiais de aprendizado tem por objetivo facilitar esse processo de escolha, de modo que os itens ofertados sejam mais atraentes aos usuários em meio à gama de opções disponíveis.

Este trabalho se propõe a apresentar uma aplicação dos sistemas de recomendação, desenvolvendo uma ferramenta para os alunos do Bacharelado em Sistemas de Informação da UNIRIO, a fim de sugerir uma grade de disciplinas a serem cursadas pelo aluno no período seguinte, a partir do seu histórico escolar. O sistema tem como base conceitos como aprendizado não supervisionado, filtragem colaborativa, e a técnica de fatoração matricial não negativa, a fim de obter um modelo baseado nas notas do aluno do curso, aprendendo características que estejam implícitas nas disciplinas do curso a fim de realizar um processo de recomendação que seja satisfatório para este domínio acadêmico.

Palavras-chave: sistemas de recomendação, aprendizado de máquina, fatoração matricial não negativa.

ABSTRACT

The crescent offer of content in the last years made choosing relevant items even more complex, motivating the emergence of recommender systems. The popularization of e-commerce and streaming services made this kind of system more present in everyday life than ever. Your application in the suggestion of movies, music, news and even learning materials aims to ease this choice process, so that the items offered are more attractive to users amidst the range of available options.

This study aims to present another application of recommender systems, developing a system for students of the UNIRIO's Bachelor's degree in Information Systems, in order to suggest subject grids to be taken in the following term. The system has unsupervised learning, collaborative filtering and non-negative matrix factorization as base concepts, in order to obtain a model based on student's grades and learning features that may be implicit in the course subjects and perform an adequate recommendation process for this academic domain.

Keywords: recommender systems, machine learning, non-negative matrix factorization.

Índice

Índice de Tabela

Índice de Figuras

Índice de Equações

Índice de Quadros

1. Introdução

1.1. Motivação

Um bom planejamento acadêmico é importante no decorrer da graduação de um aluno. De acordo com Veloso (2002) [14], a evasão de estudantes é um fenômeno complexo e comum às instituições universitárias no mundo. O planejamento inadequado da grade de disciplinas a serem cursadas em um período – em que o aluno curse muitas disciplinas difíceis simultaneamente – pode ser extremamente frustrante para o discente. É possível que um baixo índice de sucesso em um período seja um fato desmotivador e prejudique a dedicação do aluno e sua permanência na graduação.

Os sistemas de recomendação, ferramentas cujo principal objetivo é a melhor sugestão de itens relevantes aos seus usuários, têm se tornado mais populares nos últimos anos. Essa popularização está relacionada com avanços em áreas como inteligência artificial e estatística, e também devido à grande oferta de conteúdo disponível (principalmente na *web*), que torna possível a existência de ferramentas de tomada de decisão. Esses sistemas têm sido aplicados na área da educação para recomendação de materiais complementares, disciplinas e até mesmo cursos *online*. (Dascalu et al, 2016) [4].

Devido a isso, a motivação por trás deste trabalho é auxiliar os alunos do curso de Bacharelado em Sistemas de Informação (BSI) da UNIRIO no planejamento acadêmico, tentando apoiar a tomada de decisão dos estudantes utilizando um sistema de recomendação que realiza a montagem de uma grade de disciplinas baseada na identificação de afinidades dos discentes em relação às características implícitas presentes nas disciplinas (ver seção 2.3).

1.2. Objetivos

O objetivo principal deste trabalho foi o desenvolvimento de um sistema para recomendação de uma grade de disciplinas a serem cursadas em um período. Essa ferramenta deverá ser capaz de recomendar uma grade de disciplinas para o período seguinte, de alunos do Bacharelado em Sistemas de Informação da UNIRIO, com base no histórico de cada aluno, após o aprendizado de um modelo (não considerando os horários das disciplinas), utilizando dados dos históricos escolares dos alunos e a técnica de fatoração matricial não negativa empregada em alguns sistemas de recomendação.

Os objetivos secundários foram: (i) demonstrar a possibilidade da aplicação da técnica citada nos dados disponíveis dos históricos escolares dos alunos do curso; (ii) criar uma ferramenta de recomendação de grades de disciplinas funcional; e (iii) disponibilização da ferramenta para a comunidade.

1.3. Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Conceituação – Apresenta aspectos técnicos sobre sistemas de recomendação, aprendizado de máquina e a técnica de fatoração matricial não negativa utilizada na criação do algoritmo deste trabalho
- Capítulo III: Tecnologias e Ferramenta – Descreve as tecnologias utilizadas neste trabalho, assim como a ferramenta desenvolvida.
- Capítulo IV: Resultados – Demonstra resultados obtidos pela ferramenta desenvolvida. Também exemplifica o processo de recomendação proposto pelo trabalho.
- Capítulo V: Conclusão e trabalhos futuros – Reúne as considerações finais e sugere possibilidades de continuação deste trabalho.

2. Conceituação

Para a recomendação de uma grade de disciplinas para o semestre seguinte de um aluno, é necessária a obtenção de um modelo que identifique padrões relacionados às notas dos alunos e às disciplinas do curso, e que a recomendação seja feita por um algoritmo que utilize esse modelo considerando as afinidades do aluno em questão em relação às características implícitas presentes nas disciplinas, como será apresentado na seção 2.3.

Este capítulo mostra conceitos necessários para o entendimento dos sistemas de recomendação e aprendizado de máquina, além de conceitos relacionados à fatoração matricial não negativa para compreensão de como essa técnica pode ser aplicada na ferramenta proposta. Vale ressaltar que a técnica aplicada exerce um papel fundamental no problema de filtragem colaborativa.

2.1. Sistemas de Recomendação

Nos dias de hoje, a busca por conteúdo relevante é uma tarefa complicada para os usuários dada a grande oferta de produtos e a quantidade de informação disponíveis atualmente, principalmente na web, podendo tornar a busca por conteúdo relevante uma tarefa complicada para os usuários. Ao mesmo tempo, uma recomendação adequada pode ser decisiva na retenção de um usuário para uma empresa. Dados esses problemas, tornam-se necessárias ferramentas que possam auxiliar no processo de recomendação de itens e buscas dos usuários.

Os sistemas de recomendação são ferramentas de software e técnicas que têm como objetivo a sugestão mais eficaz de itens para seus usuários, de maneira que suas recomendações sejam úteis para os mesmos (Ricci et al., 2011) [11]. As sugestões podem estar relacionadas a vários tipos de domínios, como a compra de um produto em

uma loja *online*, músicas que podem ser do interesse do usuário ou até mesmo uma notícia ou página da web que o sistema julgue relevante para o mesmo.

O primeiro sistema de recomendação comercial foi denominado *Tapestry* (Goldberg et al. 1992) [5]. Com ele, os criadores também cunharam a expressão "filtragem colaborativa" que, de maneira simplificada, refere-se à necessidade do compartilhamento de dados entre múltiplos usuários a fim de obter melhores recomendações.

Assim como o aprendizado de máquina, os sistemas de recomendação também podem ser divididos de acordo com o tipo de domínio e das informações usadas para o processo de recomendação, mas, principalmente, pelo tipo algoritmo de recomendação que será usado para a realização das recomendações (Ricci et al., 2011) [11].

Os tipos de sistemas de recomendação mais citados na literatura são: baseados em conteúdo, baseados em conhecimento, demográficos, sistemas de recomendação híbridos e sistemas de filtragem colaborativa baseados em comunidade. Cada tipo será descrito brevemente, sendo os sistemas de filtragem colaborativa o foco de maior estudo para a implementação da ferramenta desenvolvida neste trabalho.

Os sistemas **baseados em conteúdo** baseiam-se em dados históricos do usuário e, em alguns casos, informações fornecidas pelos mesmos. No processo de recomendação é criado um perfil para o usuário em questão, levando em consideração as informações fornecidas (caso o sistema realize o processo de consulta) e a análise de itens previamente avaliados pelo mesmo. Com essas informações, são recomendados itens que são similares aos anteriormente avaliados pelo usuário e compatíveis com seu perfil. Sistemas que utilizam este tipo de técnica possuem esta denominação, pois o foco da análise para recomendação é o conteúdo dos itens (Herlocker, 2000) [6].

Os sistemas **baseados em conhecimento** são definidos por Burke (Burke, 2000) [3] como sistemas nos quais o processo de recomendação é feito utilizando o conhecimento estruturado em conjunto com as preferências do usuário. As recomendações são geradas ponderando o quanto a necessidade do usuário (a descrição do problema) será suprida pela recomendação (a solução para o problema em questão). É comum a aplicação desse tipo de sistemas em domínios complexos nos quais as compras de itens não são frequentes (como a compra de um carro ou apartamento, por exemplo). Uma grande vantagem dos sistemas baseados em conhecimento é a

inexistência do chamado **problema de *cold-start***, em que não é possível que sejam realizadas inferências para usuários devido à falta de informações suficientes (Ricci et al., 2011) [11].

Os sistemas de recomendação **demográficos** são os que, como o nome sugere, utilizam informações demográficas sobre os usuários para recomendações. As personalizações direcionadas aos usuários são, por muitas vezes, baseados no país ou língua. Há pouca pesquisa relacionada a este tipo de sistema, apesar da sua popularidade na área de marketing.

Os sistemas de recomendação **híbridos** são aqueles que implementam uma combinação de técnicas. Esse tipo de sistema tenta suprir as desvantagens de uma determinada técnica utilizando outra técnica em conjunto.

Os sistemas de **filtragem colaborativa** são considerados os mais populares e implementados dentre os sistemas de recomendação. Neles, as recomendações são realizadas com o auxílio das avaliações de vários usuários pertencentes a um grupo. Com base nestas avaliações, os sistemas calculam a similaridade entre usuários baseados em seus dados históricos. Schafer et al. (2001) [12] definem o processo de filtragem colaborativa como “correlação pessoa-a-pessoa”.

O maior desafio na concepção de métodos de filtragem colaborativa é relativo à esparsidade dos dados (Aggarwal et al., 2016) [1]. As matrizes que dizem respeito às avaliações geralmente são esparsas (já que os usuários avaliam apenas uma pequena parcela da grande quantidade de itens disponíveis do domínio) e, devido a isso, ocorre uma grande quantidade de avaliações não observadas.

2.2. Aprendizado de Máquina

O aprendizado de máquina é uma área de estudo da inteligência artificial. De acordo com Samuel (1959), é o campo de estudo que dá aos computadores a habilidade de aprender sem que sejam explicitamente programados. Essa área de estudo tem como base a ideia de que sistemas podem aprender com dados, identificar padrões, simular a inteligência humana e, em alguns casos, até mesmo tomar decisões com pouca ou nenhuma intervenção.

O aprendizado de máquina possibilita que sistemas consigam lidar com dados de uma maneira mais eficiente. Por vezes, é comum que a informação contida nos dados

não esteja explícita e, nesses casos, são utilizadas técnicas de aprendizado de máquina para que os dados sejam interpretados e tornem-se informações úteis. Segundo Mitchell (2006) [9], a medida que indica o progresso dessa área é a sua aplicação significativa em problemas do mundo real, como reconhecimento de fala, classificação de documentos, visão computacional, e outros.

O tipo e o método pelos quais os dados de treinamento são recebidos têm influência no tipo de tarefa a ser realizada e nos dados de teste usados para avaliar o sistema ou algoritmo (Mohri et al., 2018) [10]. As tarefas realizadas por meio de aprendizado de máquina são tradicionalmente divididas em três categorias: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço.

No aprendizado supervisionado são fornecidas entradas e saídas rotuladas e fica a cargo do algoritmo aprendiz realizar predições para os pontos ainda não observados, de modo que ele aprenda uma regra geral (uma função de inferência) que mapeie as entradas para suas respectivas saídas. Exemplos desta categoria são os problemas de classificação e regressão.

Na **classificação**, define-se que os elementos pertencem a um número finito de classes, e o objetivo é estabelecer uma regra que mapeie entradas novas para essas classes existentes (Michie et al, 1994) [8]. A figura 1 mostra um caso de rótulos e dados rotulados que são consumidos por um aprendiz, gerando um modelo e possibilitando novas previsões.

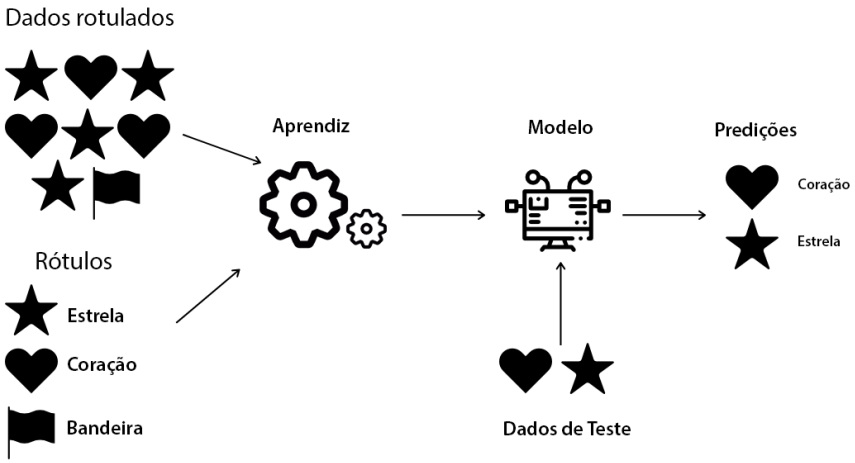


Figura 1 – Exemplo de classificação

Fonte: Autor

A **regressão** modela como certas variáveis influenciam uma variável de resposta (Y). Diferente da classificação, a regressão prediz valores numéricos, e a tarefa dos algoritmos baseados em regressão é encontrar um modelo matemático capaz de predizer valores de Y com base em novos valores de variáveis preditoras X. A medida de erro é baseada na magnitude da diferença entre o valor previsto e o valor verdadeiro (Mohri et al, 2014) [10]. A figura 2 apresenta um exemplo de regressão linear que modela o valor de um salário Y em relação aos anos de experiência X.

Figura 2 – Exemplo de regressão linear.
Fonte: https://www.analyticsvidhya.com/blog/2021/01/a-quick-overview-of-regression-algorithms-in-machine-learning/ (Acesso: jul. 2021)

No aprendizado não supervisionado não existem rótulos. Desta forma, fica a cargo do algoritmo de aprendizado extrair um padrão ou modelo dos dados. Como não existem saídas rotuladas, pode ser difícil avaliar o desempenho do algoritmo em questão. A **redução de dimensionalidade** é um exemplo desta categoria, sendo uma técnica para lidar com dados que possuem um grande número de dimensões[1]. Desta forma, o objetivo é a redução do número de dimensões de forma que a nova representação retenha propriedades relevantes dos dados originais e tente facilitar a interpretação. A técnica é utilizada em campos como o reconhecimento de voz e processamento de fotografias digitais (Van der Maaten et al, 2009) [13].

1 Cada propriedade presente nos dados pode ser considerada uma dimensão. Um dado com muitas dimensões é representado por diversas características. A redução de dimensionalidade tem por objetivo a redução dessas dimensões a fim de extrair apenas propriedades relevantes dos dados.

No aprendizado por reforço, o aprendiz interage ativamente com o ambiente para coletar informações de forma a atingir um determinado objetivo. O aprendiz tenta maximizar as recompensas que ganha ao interagir com o ambiente, e eventualmente atinge um ponto onde deve explorar o ganho de novas informações ao realizar novas ações com esse ambiente. Um exemplo dessa categoria são os carros autodirigíveis.

2.3. Fatoração Matricial Não Negativa

A filtragem colaborativa pode ser subdividida em algumas categorias: baseada em modelos, baseada em memória e de filtragem híbrida (Bokde et. al, 2015) [2].

Os métodos de fatoração matricial são fundamentados em modelos de fatores latentes[2], considerados estados da arte para a filtragem colaborativa baseada em modelos (Sammut, WEBB, 2011). Pode-se dizer que, em relação a sistemas de recomendação, os fatores são dependentes de domínio. Segundo Koren et al. (2009) [7], em um domínio relacionado a filmes, por exemplo, um fator pode ser evidente (como o gênero do filme) ou menos óbvio (como a quantidade de suspense presente no enredo). Este tipo de modelo é considerado a abordagem mais precisa quando existe grande esparsidade nos dados de um sistema recomendação (Bokde et al. 2015) [2].

A matriz R é uma matriz de tamanho |U| x |I| que contém todas as avaliações dos usuários para os itens, onde **U** é o número de usuários e I o número de itens. A Figura 3 representa um exemplo de uma matriz de avaliações hipotética:

	I1	I2	I3	I4
U1		6.0	3.8	
U2	9.0		6.0	
U3		2.0		
U4		4.5	5.0	3.1

Figura 3 – Uma matriz R de avaliações 4x4.

2 Fatores latentes são variáveis que não são diretamente observadas, mas sim inferidas através de outras variáveis observáveis. No campo da economia por exemplo, a qualidade de vida é um exemplo de fator latente: não pode ser diretamente medida, mas é inferida a partir de outros fatores.

Fonte: Autor

É necessário que sejam descobertos os fatores latentes. Para isso, encontram-se duas matrizes P (de tamanho |U| x K) e Q (de tamanho K x |I|) as quais o produto resultante é aproximadamente igual à matriz original de avaliações R, como demonstrados na equação 1 e na figura 4. Neste contexto, K é um tamanho qualquer.

$$R \approx PxQ^T = R'$$

Equação 1 – Aproximação da matriz original pela multiplicação de P e Q

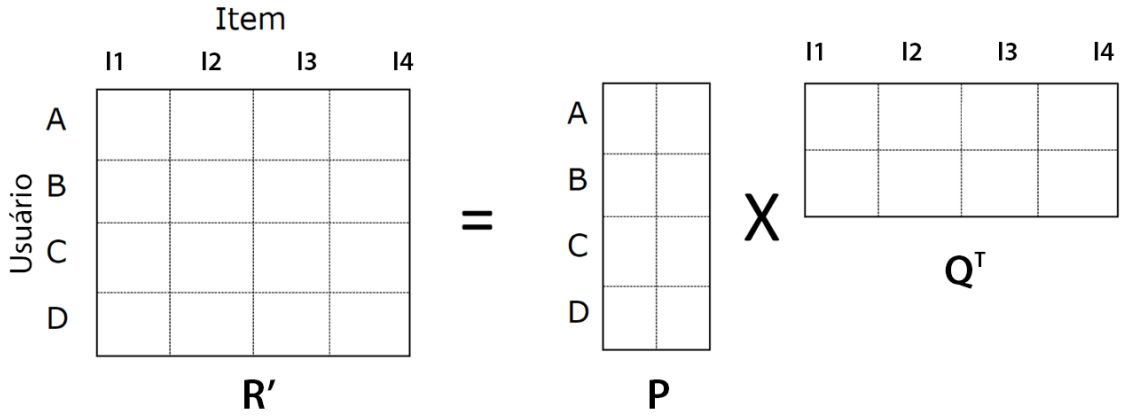


Figura 4 - Aproximação da matriz original pela multiplicação de P e Q

Fonte: Autor

Desta maneira, usuários e itens são mapeados para um espaço de dimensionalidade menor, e as interações usuário-item tornam-se produtos internos deste espaço. Sendo assim:

- Cada item **i** é associado a um vetor $q_i \in R^f$ de características; e
- Cada usuário **u** é associado a um vetor $p_u \in R^f$ de preferências.

O produto escalar resultante da equação $p_u \cdot q_i^T$ representa uma interação entre o usuário **u** e o item **i**. Mais especificamente, esse produto representa o interesse ou afinidade do usuário em relação a determinada característica do item em questão. Deste modo, ocorre um cálculo de aproximação da avaliação real do usuário para o item, representada pela equação 2.

$$p_u \cdot q_i^T = r'_{ui}$$

Equação 2 – Estimativa da avaliação do usuário

A Figura 5 é uma representação gráfica do processo, onde a tabela de usuários possui quatro entradas e suas avaliações para três itens (filmes) distintos, e os valores zerados denotam itens ainda não avaliados pelos usuários 2 e 4. A tabela P diz respeito ao quanto a característica é impactante para o usuário. A tabela Q tem relação com a presença das características nos itens. A tabela de valores previstos (o equivalente à matriz *R'*) é aproximada pelo produto das matrizes P e Q (a segunda já representada de maneira transposta no exemplo).

Tabela de Usuários			
Usuário	Filme 1	Filme 2	Filme 3
Usuário 1	5	8	3
Usuário 2	3	4	0
Usuário 3	8	7	3
Usuário 4	7	6	0

Valores previstos			
Usuário	Filme 1	Filme 2	Filme 3
Usuário 1	5.10	7.92	2.97
Usuário 2	3.07	3.94	1.54
Usuário 3	7.69	7.02	3.03
Usuário 4	6.91	6.08	2.63

Tabela P (Preferências)		
Usuário	Característica 1	Característica 2
Usuário 1	0.57	2.53
Usuário 2	0.53	1.15
Usuário 3	2.10	1.55
Usuário 4	1.83	1.34

Tabela Q (Características)			
Característica	Filme 1	Filme 2	Filme 3
Característica 1	2.75	1.24	0.69
Característica 2	1.40	2.85	1.02

Figura 5 – Exemplo de predição

Fonte: Autor

O erro do algoritmo é representado pela equação 3, e diz respeito a quão precisa foi a previsão da avaliação de um usuário para determinado item em relação à avaliação real. Ele será calculado pela subtração do valor conhecido de uma avaliação r_{ui} pelo seu valor previsto.

$$e_{ui} = r_{ui} - r'_{ui}$$

Equação 3 – Fórmula do cálculo de erro da avaliação

Em relação às entradas faltantes é sugerido que apenas as avaliações conhecidas sejam utilizadas, ao contrário de que é feito por abordagens como a decomposição de valores singulares (SVD, do inglês *Singular Value Decomposition*), que insere valores para que a matriz de avaliações torne-se densa. Vale ressaltar que a inserção de valores

pode ser custosa, já que a matriz R é tipicamente grande, e pode causar distorções nos dados reais (Koren et al, 2009) [7].

A raiz quadrática média (**RMSE**, do inglês *Root Mean Square Error*) é uma medida utilizada por sistemas de recomendação baseados em fatoração matricial não negativa para medir os erros de predição. Sendo assim, para que o aprendizado dos vetores de fatores q_i e p_u seja feito de maneira ótima, o sistema tem por objetivo a minimização dessa medida de erro. A figura 6 apresenta a fórmula da soma dos quadrados das estimativas dos erros (SSE, do inglês *sum of squared estimate of errors*) e a fórmula o RMSE.

$$\mathrm{SSE} = \sum_{(u,i)} e_{ui}^2$$

$$RMSE = \sqrt{SSE} = \sqrt{\sum (R - R')^2}$$

Figura 6 - Fórmulas da SSE e da RMSE

Fonte: Autor

Uma aplicação conhecida do método foi realizada no *Netflix Prize* ocorrido de 2006 a 2009 (Koren; Bell; Volinsky, 2009) [7]. A competição buscava a melhoria do sistema de recomendação utilizado pela plataforma, e os autores aplicaram variações do método com o objetivo de minimizar o erro das avaliações recomendadas, utilizando um dataset disponibilizado pela empresa (uma matriz composta por 500.000 usuários anônimos, 17.000 filmes e aproximadamente 100 milhões de avaliações). Os autores implementaram 5 versões da *NMF (Non-negative Matrix Factorization)*, com melhora gradual do *RMSE* em cada uma delas (Koren; Bell; Volinsky, 2009, p. 36) [7]. A Figura 7 apresenta o gráfico da acurácia atingida por cada modelo utilizado pelos autores, e também demonstra como ela aumenta proporcionalmente à dimensionalidade do modelo (se aproximando mais do número de dimensões original dos dados).

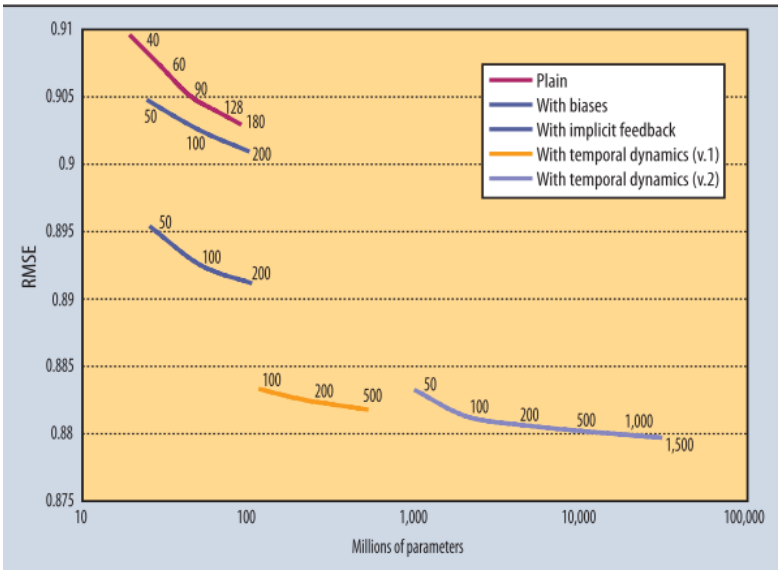


Figura 7 – Acurácia dos modelos obtidos pelos autores

Fonte: (Koren; Bell; Volinsky, 2009, p. 36)

A fatoração matricial é interessante pela sua flexibilidade, que independe do tipo dos itens presentes no domínio, podendo ser produtos a serem comprados, filmes assistidos, etc. Neste trabalho será aplicada esta técnica no domínio acadêmico na tentativa de sugerir uma grade de disciplinas a serem cursadas no período seguinte por um aluno, ver capítulo 4.

3. Tecnologias e Ferramenta

Com o objetivo de auxiliar os alunos do BSI da UNIRIO a escolherem disciplinas para montagem de grade do período seguinte, foi desenvolvida uma ferramenta de recomendação que faz uso da fatoração matricial não negativa.

Na seção 3.1 são descritas as ferramentas que foram utilizadas no desenvolvimento do código da ferramenta e, em seguida, uma descrição sobre o programa de teste e como é realizado o processo de recomendação.

3.1. Tecnologias utilizadas

Python

Python é uma linguagem de programação interpretada, de alto nível, que possui a proposta de uma sintaxe simples e legível, sendo muito utilizada na área de ciência de dados e aprendizado de máquina[3].

Python foi a linguagem escolhida para a realização deste trabalho devido a sua facilidade de prototipação e implementação, além da disponibilidade da biblioteca *scikit-learn*, que possui módulos de fatoração matricial.

PyCharm – Community

PyCharm é um ambiente de desenvolvimento específico para a linguagem Python, desenvolvido pela empresa JetBrains. Foi o ambiente escolhido para implementação do código por ser uma IDE com diversas ferramentas como *code-completion*, *type hinting* e diversas outras facilidades que auxiliam na implementação de código na linguagem, além da gratuidade da versão Community.

---

3 https://analyticsindiamag.com/why-should-you-learn-python-for-data-science/ (Acesso: jul. 2021)

Sklearn

A *scikit-learn* é uma biblioteca gratuita, composta por módulos de aprendizado de máquina para a linguagem Python. A biblioteca possui implementações de algoritmos de diversas técnicas, como classificação, regressão, clustering e outros. Foi utilizado em específico o módulo ***sklearn-decomposition***. O módulo possui algoritmos de decomposição de matrizes, como análise de componentes principais (PCA) e fatoração matricial não negativa (NMF).

O módulo da biblioteca chamado *sklearn-decomposition.NMF* possui código para implementação da técnica de fatoração matricial não negativa, utilizando duas matrizes não negativas P e Q (chamadas de matrizes W e H na documentação[4]) cujo produto matricial aproxima-se da matriz original R (chamada de matriz X). O objetivo é a otimização (minimização) da distância entre X e o produto W · H, como mostrado na seção 2.3 do capítulo anterior.

3.2. A ferramenta de recomendação

A ferramenta tem como proposta a recomendação de uma grade de disciplinas para o período seguinte de alunos do Bacharelado em Sistemas de Informação da UNIRIO, com base no histórico das disciplinas cursadas, dele e de outros discentes. A mesmo faz uso da fatoração matricial não-negativa — fornecida nele pelo módulo *sklearn-decomposition* — para o aprendizado de um modelo e, consequentemente, realizar a predição em novos dados.

A predição é calculada considerando os pesos do aluno (os pesos do vetor p_u, presentes na matriz P) em relação às características implícitas presentes nas disciplinas (os pesos do vetor q_i, presentes na matriz Q). Esses valores são obtidos na fatoração da matriz original R. O processo ocorre no método mostrado pelo quadro 2, presente na seção 3.2.3.

3.2.1. Arquitetura

A arquitetura do programa é composta por uma função principal, responsável por chamar um conjunto de funções auxiliares. Elas são as responsáveis por: Fatorar a

4 https://scikit-learn.org/stable/modules/decomposition.html#nmf

matriz e gerar a matriz de predições, analisar quais disciplinas ainda não foram cursadas pelo aluno, quais disciplinas o aluno possui pré-requisitos para cursar e, por fim, recomendar uma grade de disciplinas balanceada, em que 50% das disciplinas são consideradas “mais fáceis” para o aluno (disciplinas em que o valor previsto foi alto) e 50% consideradas “mais difíceis” (disciplinas em que o valor previsto foi mais baixo).

Abaixo, estão descritos os papéis de cada uma das funções utilizadas pela função principal, seguido de um fluxograma de execução do código:

- A função *importarArquivos* importa para a ferramenta três arquivos *csv* que servirão de entrada para o sistema: o arquivo com notas de diversos alunos para o aprendizado do modelo, o arquivo com as notas do aluno que terá a grade recomendada e o arquivo com as aprovações do aluno em questão. Vale ressaltar que as notas são anônimas, não possuindo identificação dos alunos ou o período ao qual pertencem.

- A função *aplicarModelo* aplica a fatoração na matriz original, aprendendo o modelo e gerando as matrizes P e Q (ver seção 2.3), além de calcular o produto matricial das duas matrizes resultantes, gerando assim uma matriz próxima da matriz original, mas com valores previstos (a matriz de previsão).

- A função *listarDisciplinasFaltantes* gera uma lista com as disciplinas que o aluno ainda não cursou.

- A função *listarAprovaocoesFaltantes* gera uma lista com as disciplinas que o aluno ainda não possui pré-requisitos para cursar.

- A função *filtrarArray* filtra a matriz de notas previstas, levando em consideração apenas as disciplinas que o aluno ainda não cursou e as que possui pré requisitos para cursar. Em seguida, extrai as previsões mais altas (matérias com notas previstas mais altas) e mais baixas (matérias com notas previstas mais baixas).

- A função *montarGrade* gera a grade do próximo período, considerando um determinado número de disciplinas. A quantidade padrão de disciplinas é **4**

(sendo reduzida caso o aluno possua um total de disciplinas cursáveis menor). Metade da quantidade (arredondada para cima) é composta por disciplinas com os *scores* previstos mais altos, a outra metade (arredondada para baixo) são disciplinas com os scores previstos mais baixos.

- A função *sugerirGrade* exibe a grade recomendada para o período seguinte, levando em consideração a grade de disciplinas preparada pela função *montarGrade.*

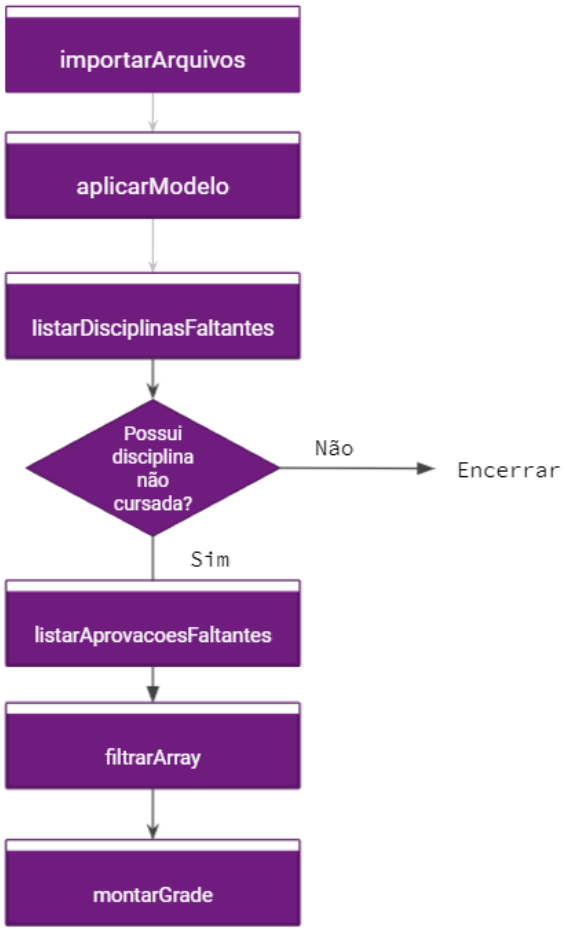


Figura 8 – Fluxograma de execução do código

3.2.2. Arquivos de entrada e limitações

O programa recebe três arquivos de entrada, que serão melhor detalhados nesta seção.

O primeiro arquivo, denominado de **base**, é um arquivo csv composto por uma matriz de notas dos alunos do curso. Este é utilizado pela ferramenta para aprendizado

do modelo baseado na matriz original. O arquivo possui 32 colunas, sendo cada coluna uma disciplina obrigatória do BSI-UNIRIO e cada linha um aluno. A Tabela 1 mostra a ordenação das disciplinas em ambos os arquivos e suas respectivas siglas, que são exibidas no momento da recomendação.

Disciplina	**Sigla**	**Posição no arquivo de entrada**
Fundamentos de Sistemas de Informação	FSI	1
Técnicas de Programação 1	TP1	2
Desenvolvimento de Páginas Web	DPW	3
Organização de Computadores	OC	4
Matemática Básica	MB	5
Técnicas de Programação 2	TP2	6
Introdução à Lógica Computacional	ILC	7
Análise Empresarial e Administrativa	AEA	8
Cálculo 1	CALC1	9
Álgebra Linear	ALG LINEAR	10
Banco de Dados 1	BD1	11
Estruturas de Dados 1	EDD1	12
Sistemas Operacionais	SO	13
Estruturas Discretas	EDC	14
Cálculo 2	CALC2	15
Probabilidade	PROB	16
Análise de Sistemas	AS	17
Estruturas de Dados 2	EDD2	18
Interação Humano-Computador	IHC	19
Redes de Computadores 1	REDES 1	20

Linguagens Formais e Autômatos	LFA	21
Estatística	ESTAT	22
Projeto e Construção de Sistemas	PCS	23
Banco de Dados 2	BD2	24
Análise de Algoritmos	AA	25
Redes de Computadores 2	REDES 2	26
Empreendedorismo	EMP	27
Programação Modular	PM	28
Projeto e Construção de Sistemas com SGBD	PCS-SGBD	29
Administração Financeira	ADM FIN	30
Processos de Software	PS	31
Gerência de Projetos de Informática	GPI	32

Tabela 1 – Disciplinas e suas posições nos arquivos de entrada da ferramenta

Vale ressaltar que a matriz possui valores desconhecidos (disciplinas ainda não cursadas pelos alunos), que são preenchidos com 0. No caso da presença de repetências nos históricos, foi feita uma mediana das notas da disciplina em relação a quantas vezes a mesma foi cursada pelo aluno.

O segundo arquivo .csv é denominado **alvo**. Este também possui 32 colunas mas apenas uma linha, estando contida nela as notas do aluno que será o alvo da recomendação, como visto na Figura 9. Caso o aluno possua repetência na disciplina, é feita mediana das notas que recebeu em relação às vezes que já cursou as disciplinas (um aluno que cursou TP1 três vezes com notas 2, 4.5 e 7 terá um valor de 4.5 para essa disciplina no arquivo alvo, por exemplo).

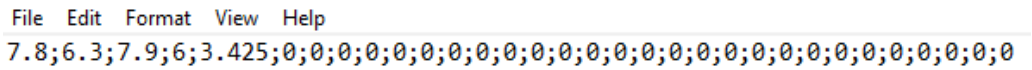


Figura 9 – Exemplo de arquivo alvo

Fonte: Autor

O terceiro arquivo .csv é o arquivo de **aprovações**. Este é similar ao arquivo **alvo**, com 32 colunas e 1 linha. O arquivo diz respeito às aprovações do aluno em questão (que não podem ser inferidas diretamente a partir das suas notas). O valor **1** para cada posição denota aprovação na disciplina, o valor **0** denota o caso contrário, como exemplificado na Figura 10.

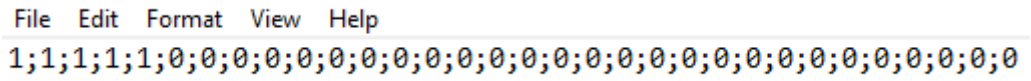


Figura 10 – Exemplo de arquivo de aprovações
Fonte: Autor

A formatação correta dos três arquivos é imprescindível para o funcionamento da ferramenta:

- O arquivo **base** não possui limitação quanto ao número de linhas, mas deve possuir as 32 colunas, com estas separadas por ponto e vírgula, e os valores decimais das entradas devem usar ponto no lugar de vírgula. A má formatação deste arquivo pode fazer com que a ferramenta não consiga realizar o passo de aprendizado do modelo e, consequentemente, não consiga realizar uma recomendação de grade de disciplinas.

- Tanto o arquivo **alvo** quanto o arquivo de **aprovações** devem possuir apenas uma linha, com as 32 posições separadas por ponto e vírgula. Os valores decimais presentes no arquivo **alvo** também devem usar ponto no lugar de vírgula. O arquivo de aprovações deve possuir apenas entradas binárias (1 para aprovado, 0 para não aprovado), como citado anteriormente.

3.2.3. Métodos do Programa de Teste

Nesta seção serão descritos os métodos responsáveis pelo funcionamento da ferramenta, são apresentados na ordem em que são chamados e seus códigos podem ser vistos no Apêndice A.

Método de importação de arquivos

Este é o primeiro método chamado pelo programa, o pseudocódigo está apresentado no Quadro 1. Sua função é importar os três arquivos *.csv* que contém, respectivamente:

- A matriz de notas dos estudantes, que é base para aprendizado do modelo;
- A matriz com 1 (uma) linha que contém o estudante a ter as disciplinas recomendadas, como apresentado na Figura 9; e
- A matriz com 1 (uma) linha que diz respeito às aprovações do estudante a ter as disciplinas recomendadas, para verificação de pré-requisitos faltantes (como apresentado na figura 10).

Importe a matriz de notas dos estudantes para uma variável
Importe a matriz de dados alvo para uma variável
Importe a matriz com as aprovações para uma variável

Saída: Matriz das notas base, Matriz dos dados alvo, Matriz das aprovações

Quadro 1 – Código responsável pela importação dos arquivos a serem utilizados pelo programa
Fonte: Autor

Método de aplicação do modelo

Este método é o responsável pelo aprendizado do modelo e sua aplicação nos dados do aluno a ter as disciplinas recomendadas, o pseudocódigo será mostrado no Quadro 2. Este chama funções do módulo *sklearn-decomposition.NMF*. O modelo utiliza 16 componentes, com uma condição de parada de no máximo 2000 iterações ou até atingir a medida de tolerância de erro padrão da biblioteca (0.0001). A função a ser minimizada (erro) pelo algoritmo é a função de erro padrão utilizada pela biblioteca (norma de Frobenius, considerada uma extensão da norma euclidiana para matrizes, medida de erro apresentada no capítulo 2), mostrada na Figura 11. O retorno do método será o produto das matrizes $P \cdot Q$, que resulta em uma matriz com as notas prevista para o aluno.

$$d_{\mathrm{Fro}}(X, Y) = \frac{1}{2} ||X - Y||_{\mathrm{Fro}}^2 = \frac{1}{2} \sum_{i,j} (X_{ij} - Y_{ij})^2$$

Figura 11 – Norma de Frobenius

Fonte: https://scikit-learn.org/stable/modules/decomposition.html#nmf (Acesso: ago. 2021)

Entrada: Matriz das notas base, matriz das notas do aluno alvo. **Inicialize** os parâmetros do modelo. **Aprenda** o modelo utilizando a matriz original. **Aplique** o modelo aprendido nos dados alvo, e salve a matriz de pesos P em uma variável. **Salve** a Matriz Q de características aprendidas em uma variável. **Retorne**: O produto escalar das matrizes P · Q.

Quadro 2 – Método de aprendizado e aplicação do modelo
Fonte: Autor

Método para listagem de disciplinas faltantes

Esse método é responsável por analisar os dados originais do aluno e obter quais disciplinas ainda não foram cursadas pelo mesmo, retornando uma lista de nomes e outra de índices. O pseudocódigo está descrito no Quadro 3.

Entrada: Matriz de notas do aluno.
Crie uma nova lista vazia, de índices de disciplinas não cursadas.
Para cada (índice, elemento) em (matriz_original) **faça:**
    Se elemento = 0:
        Insira o índice do elemento na nova lista.
Crie uma nova lista vazia, de nomes das disciplinas não cursadas.
Compare a lista que contém todas as disciplinas com a lista de índices de não cursadas:
    Insira na lista de nomes de disciplinas não cursadas todos os elementos em comum.
Saída: Lista de disciplinas não cursadas, lista de índices das disciplinas não cursadas.

Quadro 3 – Pseudocódigo do método para obtenção de disciplinas faltantes
Fonte: Autor

Caso a lista retornada de disciplinas não cursadas esteja vazia, o programa exibe a mensagem de que não será possível realizar recomendações e é encerrado.

Método para listagem de aprovações e pré-requisitos faltantes

Esse método é responsável por analisar os dados de aprovação do aluno (arquivo 3) e obter as disciplinas em que ainda não foi obtida a aprovação, além de quais disciplinas ainda não possui pré-requisitos pra cursar. O método retorna as três listas a seguir:

- Lista com nomes das disciplinas que o aluno ainda não possui aprovação;
- Lista com os índices das disciplinas que ele ainda não possui aprovação; e
- Lista com índices das disciplinas que o aluno não possui pré-requisitos pra cursar.

O objetivo é que essas listas sejam utilizadas pelo método seguinte, responsável por extrair as N maiores e melhores recomendações de disciplinas que o aluno ainda não cursou e está apto a cursar (possui os pré requisitos). O nome deste método é **listarAprovacoesFaltantes** (ver apêndice A).

Entrada: Matriz de aprovações do aluno.
Crie uma nova lista vazia, de índices de disciplinas onde o aluno ainda não foi aprovado.
Para cada índice, elemento em matriz_aprovacoes **faça:**
    Se elemento = 0:
        Insira o índice do elemento na nova lista.
Crie uma nova lista vazia, de nomes das disciplinas onde ele ainda não foi aprovado.
Compare a lista que contém todas as disciplinas com a lista de índices de não aprovações:
    Insira na lista de nomes de disciplinas não cursadas todos os elementos em comum.

Crie uma nova lista vazia, de índices de disciplinas que o aluno não possui pré requisitos.
Para cada (elemento) em (lista de índices onde o aluno não foi aprovado):
    Para cada (índice, disciplina) em (lista de pré requisitos de disciplinas) **faça:**
      Se elemento **não** possui lista de pré requisitos **então** não faça nada.
      Senão:
        Se elemento onde aluno não foi aprovado ⊂ na lista de pré requisitos da disciplina:
          Insira o índice dela na lista de disciplinas que o aluno não possui pré requisitos.

Saída: Lista de disciplinas não cursadas, lista de índices das disciplinas não cursadas, lista de disciplinas que o aluno ainda não pode cursar.

Quadro 4 – Pseudocódigo do método de aprovações faltantes
Fonte: Autor

O pseudocódigo pode ser exemplificado pelo exemplo a seguir, em conjunto com a Figura 12, que apresenta a estrutura da lista de disciplinas que possuem pré-requisitos. Cada posição corresponde a uma disciplina (seguindo a ordem da Tabela 1), e as sublistas são seus pré-requisitos. As entradas com *none* indicam que a disciplina não possui pré-requisitos:

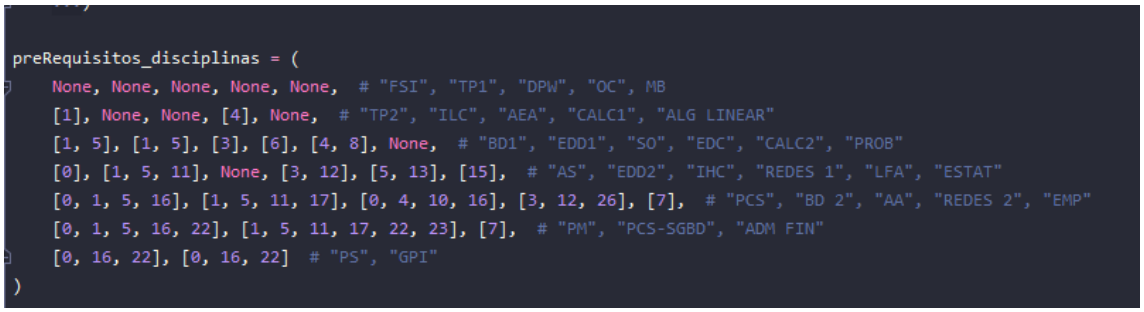
- O aluno não foi aprovado na disciplina **Técnicas de Programação 1**;
    - TP1 consta no arquivo de aprovações com 0 (não aprovado)

- Assim, a primeira parte do método insere o índice de TP1 na lista de índices de disciplinas onde o aluno não foi aprovado;

- A segunda parte do método percorre simultaneamente esta lista e a lista de disciplinas com pré-requisitos;

- Na lista de disciplinas com pré-requisitos está **TP2** (posição 6), que possui **TP1** em sua sublista de pré requisitos

    - Devido a isso, o índice de TP2 será inserido na lista de disciplinas que o aluno não possui pré-requisitos para cursar.



```
preRequisitos_disciplinas = (
    None, None, None, None, None,  # "FSI", "TP1", "DPW", "OC", MB
    [1], None, None, [4], None,  # "TP2", "ILC", "AEA", "CALC1", "ALG LINEAR"
    [1, 5], [1, 5], [3], [6], [4, 8], None,  # "BD1", "EDD1", "SO", "EDC", "CALC2", "PROB"
    [0], [1, 5, 11], None, [3, 12], [5, 13], [15],  # "AS", "EDD2", "IHC", "REDES 1", "LFA", "ESTAT"
    [0, 1, 5, 16], [1, 5, 11, 17], [0, 4, 10, 16], [3, 12, 26], [7],  # "PCS", "BD 2", "AA", "REDES 2", "EMP"
    [0, 1, 5, 16, 22], [1, 5, 11, 17, 22, 23], [7],  # "PM", "PCS-SGBD", "ADM FIN"
    [0, 16, 22], [0, 16, 22]  # "PS", "GPI"
)
```

Figura 12 – Lista de disciplinas com pré requisitos
Fonte: Autor

Método para extração de N maiores e menores recomendações do aluno

Este método é responsável por preparar as listas com as recomendações mais altas e mais baixas do aluno, que serão usadas pelo método de montagem de grade a ser recomendada. O método recebe a matriz de notas previstas do aluno, e a lista de disciplinas não cursadas e a lista de disciplinas que ele não possui pré-requisitos. Utilizando listas auxiliares, são retornadas duas sublistas, uma com os elementos onde os valores recomendados foram mais altos (“melhores” disciplinas para o aluno cursar no período seguinte) e outra onde os valores recomendados foram os menores (“piores” disciplinas, as que o aluno pode ter maior dificuldade).

Entrada: Matriz de notas previstas do aluno, lista de disciplinas não cursadas, lista de disciplinas que aluno não possui pré-requisitos.

Crie uma lista vazia de disciplinas não cursadas.
Adicione à lista apenas as disciplinas da matriz original que **também** estão presentes na lista de disciplinas não cursadas.

Crie uma lista vazia de disciplinas que o aluno possui pré requisitos para cursar.
Adicione à lista apenas as disciplinas da matriz original que **não** estão presentes na lista de disciplinas que aluno não possui pré requisitos.

Crie uma lista "filtragem final", apenas com os elementos em comum entre as listas de disciplinas não cursadas e o dicionário de disciplinas que o aluno possui pré requisitos para cursar .

Crie uma lista de tamanho **piso(**(tamanho filtragem final)/2**)** com as menores previsões para o aluno.
Crie uma lista de tamanho **teto(**(tamanho filtragem final)/2**)** com as melhores previsões para o aluno.

Saída: lista de menores previsões, lista de maiores previsões, tamanho da lista "filtragem final".

Quadro 5 – Pseudocódigo do método de extração de recomendações.

Fonte: Autor

Método para montagem da grade de disciplinas

Este método é responsável pela geração e exibição da grade de disciplinas a ser recomendada pelo aluno.

O método recebe como entrada as listas de previsões geradas, e, por padrão, recomendará 4 disciplinas a serem cursadas pelo aluno, sendo duas dessas disciplinas em que o valor da recomendação foi mais alto — o que indica que o aluno pode ter mais facilidade pra cursá-las — e duas com valores de recomendação baixo (o caso oposto). O objetivo dessa divisão é criar um período considerado balanceado.

O número de disciplinas pode ser diferente do valor padrão, seja alterando a variável *numDisciplinas* presente no método ou caso o tamanho da lista filtrada anteriormente seja menor que esse valor (ex: 3 disciplinas). No caso de um valor ímpar, o sistema recomendará o piso da metade do valor para as recomendações ruins e o teto da metade para as recomendações boas (no caso de 3 disciplinas, 2 disciplinas "boas" e 1 "ruim"). O pseudocódigo está representado no Quadro 6.

Entrada: lista de menores previsões, lista de maiores previsões, tamanho da lista "filtragem final"
Inicialize a variável (numDisciplinas) com o valor 4.
Se o tamanho da filtragem final for maior que a variável numDisciplinas:
    (Tamanho) = total da filtragem final.
Senão:
    (Tamanho) = número de disciplinas.
Crie a lista "gradeMontada", e insira nela:

Piso(**tamanho**/2) menores recomendações da lista de menores recomendações do aluno.
Teto(**tamanho**/2) maiores recomendações da lista de maiores recomendações do aluno.

Saída: Lista "gradeMontada", que contém as 2 menores e 2 maiores recomendações do aluno, ordenada.

Quadro 6 – Pseudocódigo do método de montagem de grade de disciplinas recomendadas

Fonte: Autor

Apresentados os métodos que garantem o funcionamento da ferramenta, no capítulo seguinte serão mostrados alguns resultados.

4. Premissas e Resultados

Neste capítulo são apresentados alguns resultados obtidos pela ferramenta com os dados disponíveis para teste e um exemplo de recomendação de disciplinas, assim como algumas limitações encontradas.

Para a obtenção do modelo que foi utilizado foram realizados testes utilizando dados dos históricos de 270 alunos do curso, formando uma matriz 270 x 32 com 8.640 registros. Deste total, dados de 258 alunos foram utilizados como treino para o aprendizado do modelo e dados disponíveis de apenas 12 formados como dados de teste para verificação do *RMSE* obtido pela ferramenta.

Pré-processamento

No pré-processamento dos dados foram considerados apenas alunos que possuíam pelo menos o primeiro período cursado. Essa medida foi adotada para evitar que a matriz possuísse muitos zeros de alunos que não prosseguiram com o curso (podendo ser considerados, de certa forma, usuários que não foram retidos por uma plataforma e não agregam muita informação útil para as recomendações), possivelmente influenciando os resultados obtidos no modelo.

Além disso, foi feita a mediana das notas do aluno em caso de repetência em disciplinas. Caso as repetências não fossem consideradas, não existiriam notas abaixo de **5** nos dados, não representando desta forma as possíveis dificuldades enfrentadas pelos alunos no decorrer da graduação.

Premissas

Algumas premissas foram adotadas em relação às disciplinas, são elas:

- Foram desconsideradas as disciplinas eletivas e a disciplina Técnica e Práticas Discursivas na Esfera Acadêmica, por baixa relação com a grade curricular do curso;

- Não foram levadas em consideração as disciplinas optativas, graças à sua rotatividade na grade curricular, fazendo com que muitas delas tenham pouca amostragem;

- As disciplinas de Extensão (ACE I,II,III e IV) não foram incluídas por não terem relação com outras disciplinas da grade curricular, sendo matérias de cumprimento de horas relacionados a estágios ou outras atividades extracurriculares.

  Também foram excluídas as disciplinas relacionadas ao trabalho de conclusão de curso (Projeto de Graduação I e II), por serem relacionadas ao desenvolvimento do trabalho final do aluno e não exatamente às demais disciplinas.

Inicialmente, foram feitos testes para a escolha do número de componentes do modelo. Os componentes são as características presentes nos dados. Neste contexto, eles representam relações entre as disciplinas, mas que não são necessariamente explícitas (podem indicar uma correlação entre assuntos abordados entre as disciplinas, a similaridade de carga horária entre elas, ou outro fator implícito qualquer). O trabalho, porém, não tem como foco a análise de componentes obtidos.

O objetivo foi evitar tanto um cenário de *overfitting*, em que seria gerado um modelo aplicável apenas nos dados originais, quanto um cenário de *underfitting*, em que o modelo seria simples demais e não supostamente não seria adequado para realização de previsões (gerando assim um *RMSE* muito alto). O RMSE, neste contexto, pode ser interpretado como o erro em relações a valores que já estavam presentes na matriz.

Foram testados valores entre 10 e 20 componentes, números escolhidos de maneira deliberada já que não há um consenso na melhor maneira de seleção do número

de componentes para a técnica, sendo a tentativa e erro o método comumente adotado. Um valor muito próximo do número original de componentes (32) faz com que haja alta precisão, mas não extrai nenhum fator implícito contido nos dados (já que o produto das matrizes P e Q pode ser interpretado como uma combinação linear dos vetores de P pelos coeficientes de Q) e gere *overfitting*, já um valor baixo demais gera um erro demasiadamente alto. A tabela 2 mostra os números de componentes e os valores de *RMSE* obtidos, com 5 casas decimais.

Nº de componentes	***RMSE* obtido**
10	1.40388
11	1.38428
12	1.34085
13	1.35321
14	1.31540
15	1.26957
16	1.24119
17	1.19845
18	1.18277
19	1.15987
20	1.17094

Tabela 2 – Valores de *RMSE* obtidos

Os dados fornecidos são esparsos, com a matriz de notas dos alunos tendo aproximadamente 52% dos registros com notas ainda desconhecidas, o que pode exercer influência na precisão do mesmo. Foi decidido o uso de **16** componentes para geração de recomendações da ferramenta. O programa foi testado em uma máquina com sistema operacional Windows 10, processador Intel Core i7-9700K CPU @ 3.8GHz e 16GB de RAM.

Os resultados dos testes realizados com os dados de formados foram os responsáveis pelo *RMSE* de 1.24119. As tabelas com as diferenças (valor real – previsto) entre os valores originais dos dados e os valores previstos podem ser encontradas no apêndice C.

A proximidade da diferença de 0, sinaliza que o modelo conseguiu reconstruir com uma pequena margem de erro as notas originais dos formados (em um intervalo de 0 a 10). Uma breve análise do resultado mostra que com o modelo atingido, as menores diferenças entre a matriz original e a matriz prevista são das disciplinas **AS, IHC e Empreendedorismo.** Isso pode indicar que o modelo conseguiu aprender de maneira aceitável características relacionadas a estas disciplinas e minimizar o erro ao prever valores para elas. O caso contrário ocorreu para disciplinas mais avançadas como **PS** e **GPI**, onde as diferenças entre os valores reais e previstos foi alto. É necessário reforçar que conforme a matriz de base de notas torna-se mais completa, o erro entre as notas originais e previstas tende a diminuir, fazendo com que o modelo se torne mais confiável e gere melhores recomendações já que tem mais dados para a tentativa de aprendizado de características. Também é valido pontuar que, por tratar-se de aprendizado não supervisionado, não existe "resposta certa"

Segue abaixo um exemplo para o melhor entendimento do funcionamento do processo de recomendação:

Um aluno de exemplo cursou as disciplinas 1 a 17 (numeração relativa à tabela 1), como demonstrado na ver Figura 13. O arquivo das disciplinas nas quais ele já foi aprovado está representado na Figura 14. Os valores abaixo de 5 não estão incorretos, e indicam apenas uma mediana das notas do aluno em relação às vezes que já cursou as disciplinas. Este cálculo é realizado pois num cenário onde o aluno já foi aprovado nas disciplinas, as repetências não seriam consideradas, fazendo com que não existissem "notas baixas" para nenhuma disciplina.

FSI	TP1	DPW	OC	MB	TP2	ILC	AEA	CALC1	ALG LINEAR	BD1	EDD1	SO	EDC	CALC2	PROB	AS
7.8	6.3	7.9	6	3.425	9	4.65	9	3.9	8.5	8.9	8.25	7.1	7	8.2	6.21	9.5

EDD2	IHC	REDES 1	LFA	ESTAT	PCS	BD 2	AA	REDES 2	EMP	PM	PCS-SGBD	ADM FIN	PS	GPI
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 13 - Notas obtidas pelo aluno de exemplo

Fonte: Autor

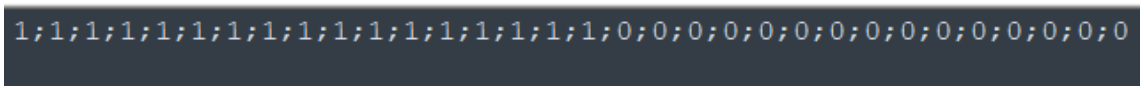


Figura 14 - Arquivo de aprovações do aluno de exemplo

Fonte: Autor

O programa de teste considera as disciplinas cursadas, indicando corretamente as matérias em que o discente ainda não foi aprovado, conforme mostra a Figura 15.

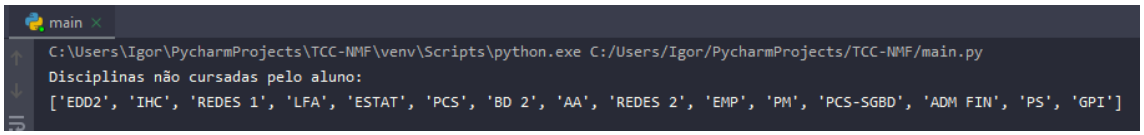


Figura 15 - Execução do programa, disciplinas não cursadas

Fonte: Autor

Após essa filtragem das disciplinas não cursadas, a ferramenta consegue indicar quais disciplinas o aluno não possui pré-requisitos para cursar, conforme indicado na Figura 16.

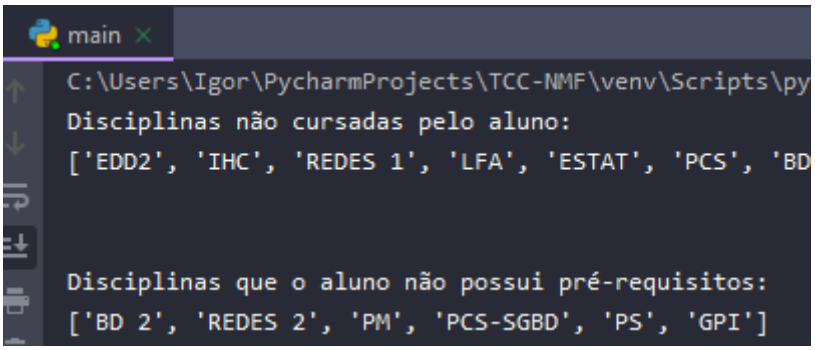


Figura 16 - Execução do programa, disciplinas sem pré requisitos

Fonte: Autor

Considerando as disciplinas que o aluno está apto a cursar, o programa extrai as maiores e menores recomendações realizadas pela técnica, apresentando-as uma tupla com um índice da disciplina (descrescido de 1 unidade, já que os índices na linguagem Python, assim como em outras linguagens, iniciam-se do 0) e o valor obtido, apenas para fins de exibição.

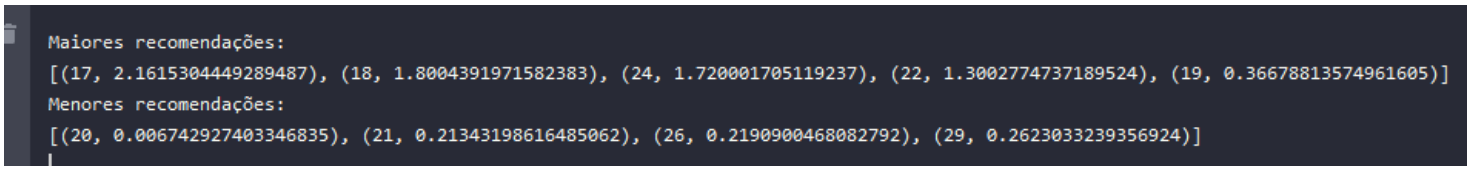


Figura 17 - Execução do programa, maiores e menores recomendações

Fonte: Autor

Por fim, o programa monta e exibe uma grade para o aluno, unindo as duas maiores recomendações (disciplinas 17 e 18, Estruturas de Dados 2 e Interação Humano-Computador) e as duas recomendações com o menor valor (disciplinas 20 e 21, Linguagens Formais e Autômatos e Estatística).

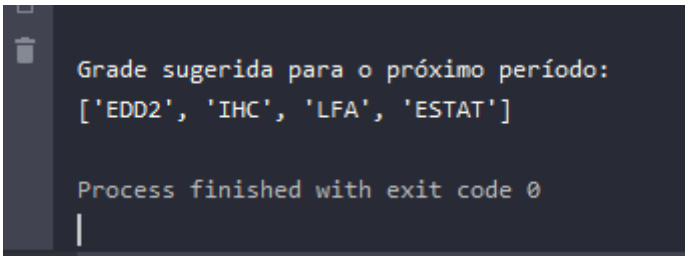


Figura 18 - Execução do programa, sugestão de grade

Fonte: Autor

Em conclusão, a ferramenta comprova sua capacidade de aprendizado de um modelo, assim como sua capacidade na sugestão de uma grade de disciplinas do BSI UNIRIO. No próximo capítulo estão apresentadas as considerações finais acerca do funcionamento da ferramenta, assim como sugestões de trabalhos futuros para a melhora dos resultados obtidos e a disponibilização da ferramenta para a comunidade.

5. Conclusão

Com o objetivo de auxiliar os alunos do BSI-UNIRIO a escolherem disciplinas para montagem da grade para a matrícula, foram estudados os conceitos relativos aos sistemas de recomendação, aprendizado de máquina e a técnica de fatoração matricial não negativa utilizada por alguns desses sistemas, e este trabalho se propôs a desenvolver uma ferramenta de recomendação com base nos estudos realizados.

A técnica foi testada sem grandes alterações nos parâmetros da bibloteca *sklearn*, com exceção do número de componentes a serem usados e o número de iterações (que não possui grande influência no resultado obtido), além de utilizar uma matriz com grande esparsidade, fazendo com que o modelo obtido ao final não tenha um erro ótimo (um *RMSE* baixo) e seja passível de melhora. Porém, ainda foi possível demonstrar o uso da técnica de fatoração matricial não negativa para a criação de um sistema simples de recomendação, como pretendido inicialmente.

Levando-se em consideração as condições não ideais dos dados disponíveis, pode-se considerar que mesmo que o resultado final fique um pouco aquém do ideal em relação ao erro do modelo, ainda assim possibilita explorações posteriores a fim de obter modelos mais completos (com maior quantidade de dados de alunos em períodos posteriores, e mais disciplinas cursadas) e aperfeiçoamento código desenvolvido. O aperfeiçoamento do modelo é um processo contínuo e, conforme os dados são enriquecidos, o erro do algoritmo na reconstrução da matriz original de notas tende a diminuir, melhorando a reconstrução da matriz original e fazendo com que as recomendações tornem-se mais confiáveis. O código da ferramenta encontra-se disponível no apêndice A deste trabalho para possibilitar sua consulta e até trabalhos futuros baseados no mesmo.

Apesar de algumas dificuldades — como o pré processamento manual da massa de dados e a incerteza inicial sobre o funcionamento da técnica nos dados disponíveis — o trabalho serviu como aprendizado em relação às diferentes técnicas e métodos de recomendação disponíveis, e também como um exemplo de como a área de recomendação e aprendizado de máquina pode ser empregada para auxiliar os discentes na sua trajetória na instituição de ensino.

Para trabalhos futuros é interessante que sejam refeitos os testes, levando-se em consideração algumas sugestões:

- Melhorar o modelo, utilizando uma matriz de notas dos estudantes que possua maior quantidade de notas dos alunos, principalmente de períodos e disciplinas mais avançadas, observando o quanto isso pode influenciar no erro de reconstrução da matriz original obtido pela técnica;

- Desenvolvimento de uma interface para criação dos arquivos alvo e de aprovação, a fim de facilitar a geração de arquivos de entrada para a ferramenta. O desenvolvimento de uma interface para o sistema também pode ser interessante (ao invés de sua operação totalmente via código e IDE);

- Analisar os componentes obtidos, a fim de entender melhor as relações entre as disciplinas encontradas pelo modelo;

- Análise da percepção dos usuários em relação a aplicação; e

- Adaptar a ferramenta (ou seu conceito) para outros cursos, testando seu desempenho quando aplicada neste tipo de cenário.

Referências Bibliográficas

[1] AGGARWAL, Charu C. et al. **Recommender systems**. Cham: Springer International Publishing, 2016.

[2] Bokde, D., Girase, S. and Mukhopadhyay, D. (2015). **Matrix Factorization model in Collaborative Filtering algorithms: A survey**. *Procedia Computer Science*, v. 49, n. 1, p. 136–146.

[3] BURKE, Robin. Knowledge-based recommender systems. **Encyclopedia of library and information systems**, v. 69, n. Supplement 32, p. 175-186, 2000.

[4] DASCALU, Maria-Iuliana et al. Educational recommender systems and their application in lifelong learning. Behaviour & information technology, v. 35, n. 4, p. 290-297, 2016.

[5] GOLDBERG, David et al. using collaborative filtering to weave an information tapestry. **Communications of the ACM**, v. 35, n. 12, p. 61-70, 1992.

[6] HERLOCKER, Jonathan Lee. **Understanding and improving automated collaborative filtering systems**. University of Minnesota, 2000.

[7] KOREN, Y., BELL, R. and VOLINSKY, C. (2009). **Matrix factorization techniques for recommender systems.** *Computer*, v. 42, n. 8, p. 30–37.

[8] MICHIE, Donald; SPIEGELHALTER, David J.; TAYLOR, Charles C. **Machine learning, neural and statistical classification**. 1994.

[9] MITCHELL, Tom Michael. **The discipline of machine learning**. Pittsburgh: Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006.

[10] MOHRI, Mehryar; ROSTAMIZADEH, Afshin; TALWALKAR, Ameet. **Foundations of machine learning**. MIT press, 2018.

[11] RICCI, Francesco; ROKACH, Lior; SHAPIRA, Bracha**. Recommender Systems Handbook**. Boston, MA: Springer US, 2011.

[12] SCHAFER, J. Ben; KONSTAN, Joseph A.; RIEDL, John. E-commerce recommendation applications. **Data mining and knowledge discovery**, v. 5, n. 1, p. 115-153, 2001.

[13] VAN DER MAATEN, Laurens et al. **Dimensionality reduction: a comparative**. J Mach Learn Res, v. 10, n. 66-71, p. 13, 2009.

[14] VELOSO, Tereza Christina MA; DE ALMEIDA, Edson Pacheco. Evasão nos cursos de graduação da Universidade Federal de Mato Grosso, campus universitário de Cuiabá–um processo de exclusão. **Série-Estudos-Periódico do Programa de Pós-Graduação em Educação da UCDB**, 2002.

APÊNDICE A – Código do Sistema de Recomendação e arquivos de exemplo

```
import math
import heapq
import numpy as np
import pandas as pd
from math import sqrt
from sklearn.decomposition import NMF
from sklearn.metrics import mean_squared_error



def importarArquivos():
    # Arquivo a ser usado de base (matriz "total")
    modeloBase = pd.read_csv(r'INSERIR CAMINHO DO ARQUIVO BASE AQUI', delimiter=';')

    # Arquivo que contém o aluno a ser previsto
    dados_alvo = pd.read_csv(r'INSERIR CAMINHO DO ARQUIVO ALVO AQUI', delimiter=';', header=None)

    # Arquivo que contém quais disciplinas o aluno já foi aprovado
    # Necessário pois não conseguimos inferir pelas notas já que é feita uma mediana das reprovações
    listaAprovacoes = pd.read_csv(r'INSERIR CAMINHO DO ARQUIVO DE APROVAÇÕES AQUI';', header=None)

    return modeloBase, dados_alvo, listaAprovacoes

def aplicarModelo(dadosBase, dadosAlvo):
    # Os parâmetros do modelo
    model = NMF(n_components=16, init=None, max_iter=5000, solver='mu', beta_loss='frobenius')

    # Prepara o modelo com a matriz original
    model.fit(dadosBase)

    # Aplica o modelo aprendido nos dados alvo
    W = model.transform(dadosAlvo)

    # Features aprendidas - A matriz das características presentes nos dados
    H = model.components_

    return np.dot(W, H)
```

```
def listarDisciplinasFaltantes(arrayBase):
    naoCursadas_idx = []
    for idx, x in np.ndenumerate(arrayBase[0:]):
        if x == 0:
            naoCursadas_idx.append(int(idx[1]))

    naoCursadas = [disciplinas[i] for i in naoCursadas_idx]

    return naoCursadas, naoCursadas_idx


def listarAprovacoesFaltantes(arrayBaseAprovacoes):
    naoAprovadas_idx = []
    for indice, d in np.ndenumerate(arrayBaseAprovacoes[0:]):
        if d == 0:
            naoAprovadas_idx.append(int(indice[1]))

    naoAprovadas = [disciplinas[i] for i in naoAprovadas_idx]

    faltamPreRequisitos = []
    for y in naoAprovadas_idx:
        for idx, x in enumerate(preRequisitos_disciplinas):
            if x is not None:
                for k in x:
                    if k == y:
                        faltamPreRequisitos.append(idx)

    faltamPreRequisitos.sort()
    faltamPreRequisitos = list(dict.fromkeys(faltamPreRequisitos))
    semReq = [disciplinas[i] for i in faltamPreRequisitos]
    print("Disciplinas que o aluno não possui pré-requisitos: ")
    print(str(semReq))

    return naoAprovadas, naoAprovadas_idx, faltamPreRequisitos


def filtrarArray(arrayOriginal, naoCursadas, semPreRequisitos):
    # Prepara duas listas, com as disciplinas que o aluno não cursou e as que ele tem pré requisitos pra cursar
    disciplinasNaoCursadas = [x for x in arrayOriginal if x[0] in naoCursadas]
```

```
    disciplinasComRequisitos = [k for k in arrayOriginal if k[0] not in semPreRequisitos]

    # Filtra com as disciplinas que ele não cursou e tem pré requisitos pra cursar
    filtragemFinal = [y for y in disciplinasNaoCursadas if y in disciplinasComRequisitos]

    menoresPrevisoes = heapq.nsmallest(math.floor(len(filtragemFinal) / 2), filtragemFinal, key=lambda x: x[1])
    maioresPrevisoes = heapq.nlargest(math.ceil(len(filtragemFinal) / 2), filtragemFinal, key=lambda x: x[1])

    return maioresPrevisoes, menoresPrevisoes, len(filtragemFinal)


def montarGrade(maioresNotas, menoresNotas, totalFiltragemFinal):
    print('Maiores recomendações: ')
    print(str(maioresNotas))
    print('Menores recomendações: ')
    print(str(menoresNotas))
    numDisciplinas = 4

    if numDisciplinas > totalFiltragemFinal:
        tamanho = totalFiltragemFinal
    else:
        tamanho = numDisciplinas

    gradeMontada = heapq.nsmallest(math.floor(tamanho / 2), menoresNotas, key=lambda x: x[1]) + heapq.nlargest(
        math.ceil(tamanho / 2), maioresNotas, key=lambda x: x[1])

    return sorted(gradeMontada, key=lambda k: k[0])


def sugerirGrade(gradeOriginal):
    gradeFormatada = []

    for i in gradeOriginal:
        for j in enumerate(disciplinas):
            if j[0] == i[0]:
                gradeFormatada += (j[1],)

    print("Grade sugerida para o próximo período:")
    print(gradeFormatada)
```

```
# Importação
base, alvo, aprovacoes = importarArquivos()
np.set_printoptions(formatter={"float_kind": lambda x: "%g" % x})

disciplinas = (
    "FSI", "TP1", "DPW", "OC", "MB", "TP2", "ILC", "AEA", "CALC1", "ALG LINEAR", "BD1", "EDD1", "SO", "EDC", "CALC2",
    "PROB", "AS", "EDD2", "IHC", "REDES 1", "LFA", "ESTAT", "PCS", "BD 2", "AA", "REDES 2", "EMP", "PM", "PCS-SGBD",
    "ADM FIN", "PS", "GPI")

preRequisitos_disciplinas = (
    None, None, None, None, None,  # "FSI", "TP1", "DPW", "OC", MB
    [1], None, None, [4], None,  # "TP2", "ILC", "AEA", "CALC1", "ALG LINEAR"
    [1, 5], [1, 5], [3], [6], [4, 8], None,  # "BD1", "EDD1", "SO", "EDC", "CALC2", "PROB"
    [0], [1, 5, 11], None, [3, 12], [5, 13], [15],  # "AS", "EDD2", "IHC", "REDES 1", "LFA", "ESTAT"
    [0, 1, 5, 16], [1, 5, 11, 17], [0, 4, 10, 16], [3, 12, 26], [7],  # "PCS", "BD 2", "AA", "REDES 2", "EMP"
    [0, 1, 5, 16, 22], [1, 5, 11, 17, 22, 23], [7],  # "PM", "PCS-SGBD", "ADM FIN"
    [0, 16, 22], [0, 16, 22]  # "PS", "GPI"
)

# Matriz com os dados a serem previstos originais
R_actual = alvo

# Matriz com os dados previstos
R_predicted = aplicarModelo(base, alvo)

# Root Mean Square Error da previsão: quanto menor, melhor
rms = sqrt(mean_squared_error(R_actual, R_predicted, squared=False))

# Cria uma lista de disciplinas não cursadas pelo aluno e uma lista com seus índices, usando as notas de entrada
listaNaoCursadas, naoCursadas_indices = listarDisciplinasFaltantes(R_actual)

if len(listaNaoCursadas) == 0:
    print("Aluno não tem mais nenhuma disciplina não cursada, não será possível realizar recomendações")
    exit()
print("Disciplinas não cursadas pelo aluno: ")
print(str(listaNaoCursadas))
```

```
# Cria uma tupla índice/disciplina para que possa ocorrer a filtragem em seguida
array = list(enumerate(R_predicted[0, :]))

listaNaoAprovadas, naoAprovadas_indices, listaSemPreRequisitos = listarAprovacoesFaltantes(aprovacoes)

# Filtra o array de previsões (considerando apenas não cursadas)
# Em seguida, extrai as previsões mais altas (matérias de maior aptidão) e mais baixas (menor aptidão)
maiores, menores, tamanhoTotal = filtrarArray(array, naoCursadas_indices, listaSemPreRequisitos)

# Monta a grade considerando um numero n de disciplinas, utilizando n/2 melhores e piores previsões
# O objetivo dessa função é criar um período "balanceado"
grade = montarGrade(maiores, menores, tamanhoTotal)
```

APÊNDICE B – Arquivos de Teste

Arquivo para aprendizado do modelo

```
8.6;7.9;9.1;8.5;6.4;9.4;0;7.2;0;0;8.5;5.1;7.77;0;0;7;8.5;2.6;7.6;0;0;1.85;0;9.2;0;0;8.5;0;0;0;0;0
9;8;8.3;9.5;5.8;7.3;4;8.1;3.9;0.6;5.9;6;7.13;3.95;3.65;7.5;6;7.1;0;0.9;0;7.2;7.69;5;5;0;10;0;1.6;0;0.725;4
2.5;0;2;0;1.65;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
10;4.5;9.5;8;4.25;2.6;5.1;9;0;0;0;0;7.7;2;0;5.09;7.69;0;8.6;0;0;7.1;0;0;0;0;0;0;8.75;0;0
9;9;9.71;7.02;7.9;9.3;6.1;10;4.75;0;9.5;8.35;9.7;8.8;0;7.28;8.56;8;10;0;0;0;0;7.6;0;0;0;0;0;9.34;0;0
6.3;0;4;0;0;1.35;2.5;0;0;6;3.4;5.2;0;4.35;5.1;3.125;2.5;5.3;8.2;4.75;4;5.35;7.64;8;5.1;8.5;4;0;0;7;5.63;0
10;0;9.83;0;9;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
7.2;4.4;9.5;9;5;4.45;4.1;0;0.15;0;6.7;1.8;1.9;0;0;5.25;2.5;0;8.4;0;0;1.55;7.13;3.65;0;0;10;0;5;7;6.9;7
9;8.1;9.75;6.18;6.8;5;6.3;10;3;7;7.3;5.66;8.9;7.5;0;0;9.05;0;10;0;2.6;0;0;0;0;0;0;0;0;0;0;0
9.5;10;10;9;9.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
9.2;6.8;8.5;8.1;7;5;3.5;7.7;2.5;4;6.6;6.2;8.4;5.5;0;5;9;5.165;9;4.85;2.1;7.3;8.62;8.7;7.6;0;9.5;5.76;7.7;8.75;8.2;7
8.7;5;8.4;8.6;10;5.7;7.5;7;8.5;8.2;6.6;6.7;7.9;8.3;8.6;9.35;6.63;8;7.2;6.1;5;8.1;8.82;8.7;6.75;7;10;6.88;7.3;7.82;0;0
7.5;6.8;9.1;7.3;6.3;0.3;7.8;9.8;0;0;0;0;5.3;0;0;6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
10;6.9;8.6;8.1;7.6;8.5;6.8;9;5.2;8.4;0;0;7.7;0;6.5;1.7;0;0;0;1.4;0;0;0;0;0;0;0;0;0;0;0;0
9.2;6.05;9.75;6.29;7.2;5;6.1;10;1.9;0;9.3;0;7.3;0;0;1.58;9.4;0;10;4.2;0;0;0;0;0;0;0;0;0;0;0;0
8.6;7.1;8.2;7.8;5.6;5.9;1.8;7.2;0;1.7;7.4;7;5.24;2.5;0;2.5;6;4.1;7.4;2.65;0;0;5.52;3.2;0;0;0;5.19;5.1;7;5.99;0
9.7;6.8;9.68;7.1;8.5;5;7.7;10;7.2;7.5;8.1;3.48;8.9;9;7.1;7.8;9.62;0;9.2;0;3.6;6.6;0;0;0;0;9;0;0;0;0;0
8.5;10;9.97;8.5;7.1;9.1;6.4;9.5;7.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
7.5;3.2;9.09;4.85;3.05;0;0;7.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
7.5;9.4;9.35;5;6.5;6.8;6.9;8.5;0;6.1;0;0;6.7;0;0;8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
9.4;0.3;5.6;5.4;2.03;0;0;5;0;0;0;0;7.4;0;0;1;7.6;0;2.2;0;0;0;0;0;0;0;0;0;0;6.5;0;0
6.5;8;9.79;1.8;2.7;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
0;0;7.98;0.2;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
8;10;10;7.5;6.8;8.3;6.3;9.8;8.1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
9;7.6;8.7;7.9;6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
9.8;7.1;8.1;8.3;7;5.1;6.1;8.5;6.6;7.3;7.7;8;8.3;7;5;6.65;9.79;6.5;8.7;7.7;0;0;7.78;7.4;0;8.8;10;0;0;7.8;0;0
9.4;4.7;9;6.6;4.75;7.9;6.8;7.5;3.4;7.7;8.2;6.38;8;5.6;0;5.43;9.74;6.8;10;4.6;0;7;8.74;9;5;0;10;0;8.8;7;8.5;0
9.2;8.4;8.9;8.6;7;8.6;5.4;7;6.8;8.8;9.6;6.7;7;9.6;7.5;7;9.5;8.5;9.5;7;7.2;8.4;9.23;8.1;0;0;9.5;0;0;8.5;0;0
9.2;8.2;9.1;8.7;7.6;9;7.9;7;8.7;8.1;9.5;7.4;8;3.8;5.8;7.35;1.43;2.9;0;2;0;0;0;0;0;0;0;0;0;0;0;0
6.5;7.6;8.89;8.1;8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
6.5;10;10;7.5;7.5;0;0;9.5;5.8;0;0;0;6.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
8.7;5;7.1;7;4.1;4;4.9;9;0;8;6.1;3.2;8.5;6.8;0;1.65;9.9;0;8.5;0;0;0;0;0;0;0;9;0;0;7.2;0;0
10;8.9;9;6.7;4.55;9;7.1;9;5.4;2;8.8;7;7.25;7.5;0;5.6;0;8.2;10;7;0;0;0;0;0;0;0;0;0;9.4;0;0
8;5.6;8.6;7.1;4.15;5.3;6.9;9.5;5;0;7.5;5.16;7.7;5.8;0;5.61;9.69;5;8.8;4.4;0;8.1;0;7.2;0;0;0;0;0;0;0;0
8.1;7.4;6.7;9;9.9;6.7;6.8;7.5;7.8;5.95;5.95;10;6.03;8.6;6.45;5.2;8;7;9.4;2.9;2.9;7;7.63;7.7;7;8.5;10;5.15;0.9;7.5;6.55;0
0;0;7.91;7.6;0;5.6;6;10;4.6;6;7.2;6.83;6.075;6.5;6.33;5;8.63;0;9.3;0;0;5.2;0;0;0;0;0;0;0;9.78;0;0
10;9.2;10;8.2;6.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
6.5;5.2;5.8;9;6.1;6.6;0.2;5.1;0.9;0.8;5.45;3.15;5.95;0;0;4.25;6;0;0;0;0;0;0;0;0;0;0;0;0;6;0;0
```

9.8;3;8.7;7.8;6.5;0;4.4;5.5;2.16;0.6;0;0;0;1.6;0;0;9.93;0;7.3;0;0;0;0;0;0;0;0;0;0;0;0;0

6;7.1;6.5;7.9;7.5;6;6.2;7;7.5;7;0;0;8.5;8;6.1;7.7;7.1;7.2;0;0;0;0;7.85;0;0;0;0;6.41;0;0;8.2;8.1

9.8;8.7;8.9;7.6;10;5.7;7.3;7.75;7.6;10;8;2.61;8.9;7;9.5;6.75;9.74;0;9.5;0;0;0;0;0;0;0;0;0;8;0;0

9.5;8.1;8.5;7.7;4.65;7.1;4.95;7;0;5.3;8.7;8.6;6.5;8.7;0;9.5;9;7.6;10;4.3;6.2;7.2;8.6;8.2;6.8;0;10;0;0;8.25;0;0

9.5;9.7;9.3;9.5;10;10;6.6;0;10;3.7;1.9;7.3;1.58;7.6;9.6;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

10;10;10;7.8;6.4;8.8;7;9;5.8;0;0;0;8.2;0;0;0;0;0;0;0;0;0;0;0;6.1;0;0;0;0;0;0;0

7.8;0;5.2;1.5;0;0;0.7;0;0;0;0;0;0;0;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0

4.9;4.3;4.4;5.5;5.15;4.3;5.2;7.2;5;6.8;7.5;5.95;5.3;8.4;5;7.2;8;6.5;10;0;0;1.6;7.04;0;0;0;0;6.21;0;7;0;0

9.7;6.15;9.1;7.4;3.7;7.8;7;8.5;0;0;6.7;3.08;9.5;2.3;0;6.11;9.2;0;8.6;5.9;0;5.3;0;0;0;0;7.5;0;0;0;0;0

9.2;7.1;9;9.3;5.25;5.2;5.2;8.5;3.9;5;6.4;2.5;6.01;5.6;5.2;7;8.9;3.77;7.8;1.8;7.2;2.5;7.04;5;6.9;0;10;0;6.1;7;0;5

1;10;7;5.28;0;7.25;7.1;0;1.615;4.1;0;0;8.1;0;0;5;0;0;0;0;0;0;0;0;0;0;0;0;0;9.12;0;0

0;0;9;0;0;0;7.7;8.8;0;6.4;9;0;8;7.1;7.2;9.75;7;0;7.6;7.1;8.6;4.95;0;7.3;7.1;9.8;10;8.2;1.95;9.25;9.7;7.7

9.8;10;8.5;8.5;7.5;9.4;8.3;9.8;5.1;7;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9;8.8;10;7.8;6.1;9.1;0;0;2.3;0;0;4.2;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

5.2;0;6.4;8.3;1.05;0;2.95;7.2;0.5;3.75;1.9;0;5.58;0;0;5.1;0;0;9.5;2.2;0;0;0;0;0;0;0;0;0;5.5;0;0

8;6.4;8.67;6.3;1.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

4.75;6.1;5.7;8.9;4;5.8;7.1;9;5.3;8;6.8;7.7;9.5;8.3;10;6.5;9.71;8.5;8.5;8.3;5.7;8.8;9.18;0;8;0;0;0;0;7.5;0;0

9.6;8;9.5;9.1;8.7;2.1;7.7;8.2;0;0;0;0;8.3;0;0;7.3;0;0;9.2;0;0;8;0;0;0;0;0;0;0;9.4;0;0

7;9.7;9.75;5.35;6;8;7.7;7;9;0;9.5;8.3;7.5;7.4;8.05;5.4;0;0;8.5;7.4;0;0;0;0;0;0;0;0;0;0;0

0;0;8.9;0;8.4;0;9.8;8.8;4.65;5.25;0;0;0;0;0;8.7;8;0;9.3;9.5;10;8.4;7.32;9;0;5.1;10;6.5;6.6;7;4.75;7.2

4.2;9;0;3.6;9.65;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.6;6.9;6.1;7.7;3.9;4.1;2.3;6;0;0;0;0;8;0;0;1.6;1.7;0;7.9;0;0;0;0;0;0;0;0;0;0;0;0

5.5;7.8;9.94;0;8.9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;7.5;0;0;0;8;9;0;1.4;7.4;8.17;6.6;7;0;8.5;9;6.6;9.6;7.8;7.2;9;10;8.9;8.7;6.3;10;8;8.7;8;8.8;7.7

9;6.15;7.6;7.9;8.6;8.5;7.2;8.5;5;8.5;7.7;5.03;9;7.4;8.5;7;9.9;8.7;9.3;0;4.05;8;8.54;7.5;0;0;0;6.94;0;8.5;9.5;0

9.1;7.8;9;8.8;7.1;8.6;5.3;7;6.8;6.8;7.9;7.4;7.13;9.4;5;7.1;7.5;7;10;4.75;7.8;6;0;6.7;6.1;8.8;10;7.4;0;7.2;8.3;7.7

0;0;0;0;0;0;7;9;0;0;0;0;0;0;0;9.13;9;0;10;9.1;0;8.6;0;7.4;8.4;8.7;10;8.1;9.2;8.8;8.7;7.7

9.3;5.7;9.69;8.13;7.3;7.5;6.7;0;7.83;6.5;7.38;4.61;8.8;5.2;0;3.8;9.8;0;0;0;0;0;0;0;0;0;0;0;0;0

8.5;6.7;8.41;7.45;6.6;7;6.1;10;7.1;9;8.5;7.34;6.6;7.2;8;7;8.56;7.4;10;6.5;6.1;0;0;0;0;0;0;0;9.56;0;0

0;0;8.5;0;6.25;5.7;6.4;9;1.835;0;8.3;0;8.3;0;0;5.17;7.7;0;7.5;1.9;0;7;0;0;0;0;9;0;0;9.78;0;0

8.5;2.4;6.6;2.1;3.45;0;4.1;8.75;0;2.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

5.95;4.5;9.5;8.8;3.5;0;3.7;5.2;1.6;1.65;0;0;8.3;0;0;0;9;0;6.2;0;0;0;0;0;0;0;0;0;0;8.25;0;0

9;1.85;8.1;6.3;0.6;0;1.3;5.8;0;0.35;0;0;4.975;0;0;0;7.4;0;7.2;3.55;0;0;0;0;0;0;8;0;0;5;0;0

0;0;0;0;0;0;0;0;0;0;0;0;9.4;0;0;0;0;0;0;0;0;0;0;0;8;0;0;0;0;0;0;0

8.5;10;9.8;5.46;0.8;4.8;3.9;10;0;0;0;0;0;7.4;0;0;0;0;0;10;0;0;0;0;0;0;0;10;0;0;0;0;0

9.5;10;9.5;0;6.2;6.25;5.5;10;7.8;0;0;0;0;6.2;0.55;2.7;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.5;9.1;8.4;8.5;7.5;6;4.9;7.2;5;5.2;8.6;6.8;5.2;8.2;5;5.6;9;8;10;5.1;7;6.6;9.85;7.6;6.8;0;9;0;0;8.5;0;0

9.4;8.6;9.6;0;5.3;7.7;5.2;7.8;5;5.9;7.4;5.5;6.35;7.4;5;6.8;8;8.5;8.8;0;5.5;6.3;7;8.2;7.3;0;9;4.89;0;7.25;4.59;0

8.3;5;8;6.7;3.05;3.05;3.8;7.2;3.9;0;3.82;0;7.9;4.5;0;4.4;6.4;0;5.9;6;0;0;0;0;0;7.2;9;0;0;0;0;0

10;10;10;6;8;9.1;8.5;9.8;8.7;8.2;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

6;0;0;0;0;0;5.8;0;4.5;3.35;0;0;0;7;6.5;3.7;9.5;7.2;0;0;2.5;0;0;6.1;6;0;0;0;0;9.56;0;0

0;0;8.5;0;0;0;7;7.5;0;6.3;8.3;5.3;6.13;6.8;0.7;1.6;7.5;5.3;8.7;7.4;0;0;0;5;5.5;3.7;10;8;0;9;5.7;2.7

0;0;7.1;0;0;0;0;0;0;0;7.6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;8;6;0;0;0;0

9.8;5.3;9.4;5.85;4;7.1;8;8.5;0;8;8.1;6;8.5;6.8;0;7.2;9.69;8.5;8.4;8.1;2.8;8.9;8.95;8.7;0;0;9;0;0;10;9.7;0

9.5;10;10;7.5;9.2;10;8.2;9.8;7.3;0;0;9.1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.5;5.8;10;7.1;5.3;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

7.4;7.4;7.5;9;6.7;7;3.7;7.2;4.35;2;7.7;7.7;4.215;0;0;5;6;2.5;7.5;2.2;0;0;2.34;0;0;0;10;0;0;7;0;0

8.7;6.2;8.7;6.6;3.9;5;6;7.8;5.9;5.8;6.8;5.14;6.415;2.2;0;2.35;9.4;3.6;4;2.2;0;0;0;0;0;0;0;0;0;8.65;0;0

0;0;7.6;0;0;0;0;3.9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

10;9.7;7.4;9.8;7;7.5;6.2;7.2;6.4;7;8.4;6.6;7.8;7;6.3;7.8;9.5;7.4;0.9;3;0;6.2;5.23;6.1;5;9.5;7;7.8;0;8.25;0;0.3

7;5;10;8.5;2.5;3.05;4.1;9.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8;6;10;6.5;8.7;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.4;4.7;8;7.4;0.8;4.4;3.25;7.8;0;1.2;6.64;0;7;1.7;0;2;9;0;7.5;0;0;0;7.83;0;0;0;10;0;0;8.25;8.5;0

8.8;5.9;8.9;6.4;4.25;6.8;5.5;7.5;2.57;5.5;5.7;6.4;8.3;4;0;5.25;9.5;6.1;7.7;3;0;7.9;6.105;7.3;6;0;0;5.23;5.3;0;8.5;6

10;8.7;9.85;5;8.4;10;8.2;9;7.1;0;0;8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.3;7.4;6.8;9;6.4;6.4;1.2;7.2;5.9;1.7;8.7;6.3;5.06;0;0;2.665;5;4.4;0;1.1;0;0;4.45;8;4.65;8.5;0;0;0;7.2;0;0

0;0;9.36;0;0;0;0;10;0;0;0;0;9.1;0;0;0;10;0;9.2;8.8;0;8.5;0;9.8;0;10;0;7.8;9.6;0;7.1;0

0;0;5;0;0;0;0;8;0;0;0;0;0;0;5.3;0;0;0;2.6;0;0;0;0;0;0;0;0;0;0;0;0;0

8.5;7;7.5;7.7;2.9;0;0.7;8.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.5;4.25;9.2;9;5;5.85;5.3;8.5;2.315;0;7.68;5.01;8.3;5.5;0;6.82;9.4;0;5.7;0;0;8;0;0;0;0;9.5;0;0;8.75;0;0

8.8;8.3;8.64;7.07;7.3;9.2;6.1;10;5.05;0;7.9;6.4;7.5;8.3;0;5.2;9.41;7.5;9.2;3.3;5;5;0;0;0;0;0;0;0;0;0;0

7.5;10;9.55;6.55;6.1;9;7.3;7;0;0;0;5.1;7.3;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

10;8.7;9.3;7.2;5.5;9.4;7.9;8.5;7.7;8;8.5;8.83;7;8.3;9.5;7.1;9;8.5;10;8.8;0;7.8;9.85;0;0;9;0;0;0;0;0;0

7.9;5.65;7.75;4.285;5;2.9;2.85;10;2.5;0;0;0;7.9;4;0;0.9;0;0;10;2.3;0;0;0;0;0;0;0;0;0;0;0;0

9.5;7.6;8;5;0;3.7;1.8;7;5;7.1;0;0;9.5;4.4;5.5;2.4;7.5;0;9.3;1.95;0;0;0;0;0;0;0;0;0;8;0;0

0;0;6.5;0;0;5.2;0;7;0;0;0;0;8.5;0;0;0;8.5;0;9.7;8.9;0;0;0;7.6;0;8.8;10;8.5;6.2;8.5;10;7.2

10;8.8;9.2;8.2;3.6;6;6.3;8.5;0;2.5;8.5;6.9;8.9;5.9;0;0;8.7;6.6;7.9;0;0;0;8.35;8.3;0;8.8;7.5;0;0;6.2;0;0

9.7;6.8;9.4;8.2;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.7;4;8.5;5;0;0;1.1;0;0;0;0;0;7.6;0;0;0;9.4;0;9.2;1.7;0;0;0;0;0;0;0;0;0;0;0;0

0;0;8.2;0;0;0;8.7;5.5;0;8.1;7.2;6;8;0;6.2;6.1;9;5.7;6.6;8.1;8.3;3.9;0;6.1;5;9.8;9;8;2.6;7;6.6;6

9.5;7.9;10;9.3;8.7;9.3;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.8;4.05;6.7;7.4;2.75;5;4.5;7.75;0;4;5.42;0;8.6;3;0;0;7;0;8.8;0;0;0;7.18;0;0;0;10;6.15;0;8.2;7;0

7.9;8.5;8;8.2;8;6.3;8;7.5;5.7;6.8;7.6;7.2;6;8.4;5.2;5.3;7.5;8;10;4.3;7.1;4.3;7.01;7;5.6;8.8;10;7.2;5.1;8.5;7.7;5.3

7;9.3;9.5;6.45;4;7.9;8.6;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.5;6.35;8.6;7.5;2.9;0.4;2.7;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

7.5;2.1;7;5.6;5.2;5.45;1.9;7.5;0;0;0;0;4.8;0;0;0.55;0;0;0;0;0;0;0;0;0;0;9;0;0;0;0;0

9.7;8.6;8.8;8.7;6.4;9.5;7.2;8;8;10;9.2;8.2;8.7;7.5;8.5;10;9.56;7.2;9.1;9.3;8.2;7.2;8.9;8.2;8.5;8;10;7.37;8.8;10;7.39;0

9;5;8.4;6.7;5.6;3.85;7.6;8;3.1;9.3;5.1;5.13;8;7.7;0;5.53;6.8;7;9.3;0;6.8;7.7;7.38;8.9;0;0;7.5;7.27;0;0;0;0

8.9;8.9;9.3;7.38;7.7;7.2;2.2;7.5;3.43;0;8.1;7;0;0;0;0;9.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

5.2;3;5.17;0;3.5;5.3;1.6;10;0;0;0;0;0;0;0;1.2;0;0;0;0;0;0;0;0;0;8.8;0;0;0;0;0;0

7.9;7.5;8.82;7.23;7.4;8.6;6.7;10;5.385;0;8.2;7.7;9.7;7.2;0;8.8;8.74;8.6;9.2;7.6;6.2;5;0;0;0;0;0;0;0;0;0;0

0;0;8.72;3.4;6.6;7.8;6.2;0;0;0;0;0;0;0;0;7.96;0;8.1;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.5;3.75;0;8.3;4;7;5.25;8;0;7;7.3;5.465;7.4;9.1;0;7.73;8.1;8.1;9.2;7.1;6.2;7.1;8.35;9.5;0;0;10;0;0;7.5;9.5;0

9.4;8.3;8.7;8.5;9.1;8.5;6.8;7;7.1;9;9.1;7.8;8.7;7.7;9.5;7.5;9.56;6.9;9.1;7.8;8.2;8.1;8.07;8;7;8;10;8.17;8.7;10;7.17;0

8.5;6.9;9.23;6.81;7.5;6.1;7.4;10;7.1;7.4;8.1;5;8.1;6.2;6.5;5.8;9.8;6;9;3.7;0;0;0;0;0;0;0;0;9.56;0;0

8.5;9.1;8;9.8;0;8.9;8.9;7;2.96;9.3;9.8;8.8;9;7.9;0;8.9;9.5;8.6;8.7;5.8;7.3;5.7;5.53;7.7;6.5;9.5;9;8.7;8.2;8.5;8.1;8.1

10;7.8;9;8.2;8.5;5.2;7.6;8.25;7;9;0;0;0;0;0;0;0;0;9.5;0;0;0;0;0;0;0;0;0;0;0

5.5;5.2;6;4.425;3.8;5.1;6.2;7.5;0;0;6.6;0;7.8;0;0;0.4;7;0;3.2;1.6;0;0;1.47;0;0;0;0;0;0;7;0;0

7.5;10;10;8.5;6.6;7;5.2;0;0.8;0;0;3.4;8.6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.4;8.9;9.8;9.4;7.1;9.1;7;9.2;7;9;9.9;8.3;9;8.3;7.5;7.1;9.5;8.5;9.4;0;7.7;7.7;9.13;8.1;8.3;0;10;8.34;0;9.25;7.97;7.9

6.3;5.45;6.6;2.385;0;0;2.4;7;0;0;0;0;0;0;0;0;0;3.3;0;0;0;0;0;0;0;0;0;0;0;0

8.1;5.4;8;9.1;8.3;4.6;4.2;6.1;8.5;0;0;0;6.59;0;5.5;7.4;0;0;0;0;0;5.4;0;0;0;0;8.5;0;0;0;0;0

8.5;7.9;9.41;7.3;8.4;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.7;7.2;10;7.2;7.6;7.6;7.7;10;6;8.5;7.02;5;9.6;9;7.5;9;8.98;0;10;0;3.7;9.2;0;0;0;0;9;0;0;0;0;0

7.4;5.5;9.76;6;7.1;3.9;6.2;9.8;3.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;7;0;8;0;7.1;5.475;8.8;8;6.9;5.7;5.6;6.8;6;9.9;9.5;7.1;0;0;4.7;7;7.28;3.1;3.5;0;0;0;0;7;0;0

9;8.9;9.86;8.8;0;8.7;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

7.6;6.05;7.97;7.17;6.6;3.3;6.8;10;3.8;4;0;0;6.8;5.3;0;1.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

10;7.6;10;7.1;8.3;9;8.4;9;0;0;8.5;7.27;8;8.9;0;0;9.57;0;10;7.4;0;0;0;0;0;0;0;0;8.91;0;0

8.5;7.3;8.82;7.55;7.8;9.7;6.1;10;5.765;0;7.8;7.06;9.1;7.2;0;8;9.35;8.8;9.2;6.4;5;5;0;0;0;0;0;0;0;0;0;0

7.5;0;10;0;0;7.3;8.3;0;0;0;0;0;9.3;0;0;8.9;0;0;0;0;0;0;0;0;0;0;7.5;0;0;0;0;0

10;2.9;8.8;7;0;0;0;7.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.2;0;8.5;7.4;0;0;3.8;7.3;0;0;5.2;5.4;7;5;0;6;7;1.75;9.5;4.15;1.8;1.7;7.54;6.8;7.9;0;10;6.51;8.9;8.25;9.7;7.3

8;6.8;7.85;6.7;6.1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.1;9.8;8.5;9.1;10;9.5;8.6;7;0;7.6;7.7;9.1;8.33;0;0;8;9;8.3;9.5;9;9.3;0;5.46;6.3;6.5;6.5;10;6.2;3.15;7;7.4;7.3

7.7;10;9.96;7.5;8.3;8.8;5.3;9.8;7;0;0;0;0;0;0;0;0;6.8;0;0;0;0;0;0;0;0;0;0;0;0

8.1;0;7.9;0;0;5.6;5.5;7.2;1.5;0;7.9;5.05;8.1;6.2;0;7;9.4;6.1;8.8;5.5;0;5.1;7.55;5.4;0;0;0;0;0;8;0;0

9.4;10;9.76;8.5;8.5;10;8.7;9.8;0;0;0;0;8.1;0;0;9.2;0;0;10;0;0;0;0;0;0;0;0;0;0;0;0;0

8.7;7.5;8.5;8.5;6.4;6.7;5;7;6.4;5.1;8.7;7.7;6.43;6.4;5.6;8.75;8.5;5;8.8;6.25;6;7.4;7.86;7.7;7.75;7.9;10;7.2;7.2;8.75;9;7.7

0;0;0;0;7;7.7;6.1;0;7.8;1.3;0;0;0;6;6;6.8;9;6;8.2;5.8;7.2;3.85;7.84;8.1;6;7.3;8.5;7.2;6.6;8.4;8.3;1.7

9;5;8.5;7;6.5;7.7;5.9;8.2;7.1;5.3;9;8.7;6.67;5.4;4.2;8;9;5.6;8.5;7.9;7;8.2;7.23;8.5;7;8.8;10;7.8;8;9.8;8.5;7.2

6;0;8.7;0;0;0;5.1;5;0;1.4;9.1;0;6.25;0;0;3.1;9.7;0;8.7;7.6;8.3;0;0;1.55;0;0;10;0;0;6.5;2.5;0

9;7.4;9.1;8.6;4.95;7.6;5.9;9.5;0.17;0;7.4;3.955;7.8;0;0;6.3;7.69;0;0;0;0;7.4;0;0;0;0;9.5;0;0;6.92;0;0

7;7;10;8.5;2.5;4.5;3.6;9.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;8;0;0;8.1;8.4;9;0;9.1;8.2;6.7;6.8;7.3;0;6.2;9.51;6;9.2;5.3;0;0;5.98;7.2;0;0;0;7.93;0;0;0;0

8;4.8;8.68;3.5;0.25;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.6;5.2;9.23;5.77;6;5.3;5.3;10;5.9;0;5.02;0;8.2;6.3;0;7.7;9.8;0;9.2;0;0;7;0;0;0;0;0;0;0;10;0;0

9.7;7.2;9.68;7.06;7.5;5;6.1;10;6.1;9.8;9.6;7.7;9.6;9.3;10;8.12;9.65;8.1;10;9.4;7.7;8.2;0;0;0;0;0;0;0;0;0;0

9.3;8.5;9.85;5;0;0;0;9;0;0;0;0;7.9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.9;8.5;9.4;9.2;7.6;8.5;7.5;8.5;8;8.5;8.1;7.88;8.8;7.9;8.4;7.25;9.4;7.5;8.7;8.1;6.5;8.6;0;9.6;0;8.5;9.5;0;0;10;0;0

8.6;0;8.8;0;0;0;6.5;7.5;0;7.2;5.5;6.9;7.3;7.3;0;4.35;7;7.6;8.4;5;0;7.3;8.4;6.6;0;8.7;9.5;0;0;0;0;0

9.4;6.95;9.69;7.38;6.4;6.3;6.1;10;4.335;1;7.92;0;8.6;9.9;0;5.16;9.8;0;10;7;0;6;0;0;0;0;0;0;0;0;0;0

9.1;9.2;9.1;8.2;5.3;9.1;6.45;7.5;7.3;2;7.6;6;6.93;7.3;2.2;6.3;7;7;8.6;7.5;6.8;5.6;8.07;7.1;5.4;6;8.5;8.6;6.1;6.25;9.1;6.8

8.5;9.3;9.9;5.3;6.6;9.8;9.2;8.5;0;0;9.68;8.67;9.6;8.1;0;10;0;0;0;0;0;0;0;0;0;7.5;0;0;0;0;0

0;0;8.1;0;6.6;0;7.4;8.4;1.8;1.2;6.5;3.35;7.57;0;0;2;0;3.45;0;7.7;9.7;0;0;7;0;5;0;4.4;1.1;7.2;0;0

10;5.3;9.2;9.6;0;7.4;6.3;9.5;0;9;8.4;5.55;9.1;7.7;0;0;10;8.7;9.7;9.7;0;0;8.38;9.6;0;10;9;7.46;8.8;0;9.7;0

0;0;5.79;0;0;0;7.4;0;0;7.5;0;0;8.3;5.5;0;6.3;0;0;0;4.5;0;0;0;0;0;0;0;0;0;0;0

8.3;2;8;7;7.2;0;0;5.5;0.55;0.5;0;0;3.7;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

10;8.2;10;7.2;7;7.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.5;0;9.2;0;0;0;0;8.3;0;0;0;0;7.4;0;0;8.5;8.5;0;0;5.8;5;8.6;10;0;0;9.7;10;8;8.5;7.2;8.1;8.4

8.6;8.2;8.8;7;7.3;7.4;6.7;8;7.8;9.3;10;10;7.23;8.2;7.1;9.5;9.5;7.5;9.6;8.4;7.9;8.3;7.64;9.4;7.9;8.8;10;7.6;8.1;9.5;8.3;7.9

8.5;8.1;9;7.7;7.1;7.3;6.7;8;6.1;0;8.6;7;8.4;9.3;8.8;6.3;9.5;7.4;8.6;6.3;5.3;0;0;9.4;0;8.3;9;0;0;8.2;0;0

7.5;9.3;9.08;9.8;5.35;6.5;6.9;10;0.5;0;0;0;8.2;0;0;0;0;0;9.4;0;0;0;0;0;0;0;0;0;9.34;0;0

7.5;9;9.76;6;8.3;8.5;6.2;9.5;5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.1;6.8;8.4;8.3;3.1;6.6;2.9;7.4;0;4;8.1;4.3;6.97;3.3;0;0;7;8.4;0;3.2;0;0;7.96;8.4;7.1;0;10;0;7;8.8;6.6;0

9.1;7.5;9;8.4;7;6.15;5.15;7.2;3.6;6.5;7.2;10;7.23;5.1;0;5.2;7.5;5.15;9;5.7;2.3;7.6;7.33;8.3;7.9;0;0;6.75;8.9;8.75;7.4;7.3

10;10;9.83;9.2;7.3;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.6;9.4;9.7;7;9.3;9.6;8.2;9;9.7;9.6;9.7;10;8;9;10;10;9.5;8.4;9.4;8.2;9.5;9.5;9.29;9.2;8.85;8.8;10;10;9.7;8.8;9.5;8.9

8.3;3.55;8.6;7.5;1.75;4.75;5;6.5;2.535;2.8;5.68;3.465;5.03;5.2;0;4.62;9;0;9;3.7;2.4;0;7.08;0;0;8.5;10;0;0;7.2;10;7

9.3;5.1;8.8;8;4.4;5.4;5.9;7;0;0;7;5.61;8.6;6.1;0;6.34;9.66;6.5;10;3.85;0;5.2;7.51;7;0;0;0;7.07;0;7.5;5.45;0

9.3;9.6;9.69;8.75;9.2;10;6.8;10;6.95;8.9;8.5;7.76;9.3;7.6;0;7.7;9.8;7.7;0;7.5;0;0;9.33;0;0;0;0;0;0;0;0;0

9.7;6.5;9.2;8.6;5;5.3;6.5;8.75;5.82;7;8.2;7.5;8.7;7.7;0;5;9.6;6.9;9.1;5.6;4.7;8.1;8.35;7;6.65;8.8;0;0;0;8;9.5;0

4.7;5;5;8;5.4;4.25;5.05;7.8;3.06;6;7.4;2.8;4.6;7.2;0;3.9;8.5;0;7.8;0;2;2.85;5.78;5.3;0;0;8.5;0;0;7.5;10;5.5

9;8.3;8.89;7.7;8.6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

5.9;3;10;9;2;0;2.2;6.2;0;0;0;0;3.615;0;0;3.25;5.25;0;8.1;1.7;0;0;0;0;0;0;10;0;0;6;0;0

8;7.3;9.65;7.1;6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.6;5.1;9.1;6.8;5.05;6.1;6.5;7.5;0;8.1;6.4;4.17;8.5;7.4;0;5.75;9.5;0;8.6;0;3.9;5;0;0;0;0;0;0;0;7.82;0;0

8.5;10;10;7.5;7.5;7.9;5.5;9.8;9.3;5.5;0;5;0;0;0;0;0;0;6.2;0;0;0;0;0;0;0;0;0;0;0;0;0

5.5;6.4;7.89;8.5;0.6;5.2;2.7;9.8;0;0;0;0;6.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

6;7.5;9.42;6;5;7.5;5.3;9.8;2.4;2.9;0;2.4;5.9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;8;5;7.5;8.1;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

7;10;5.875;9.8;0;5;6.4;9;0;7;5.62;0;7.6;4;0;5.7;0;0;8.2;0;0;0;0;0;0;0;10;0;0;9.78;0;0

10;10;10;9.2;6.6;9.1;9;9;9.8;8.5;8.92;9.37;8.5;9;0;9.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.6;5.45;8.2;7.5;0;0;4.8;8;0;6.3;0;0;5.285;0;0;4.225;0;0;0;6.3;0;0;0;0;0;0;0;0;0;0;0

9.2;0;7.3;5;0;8.9;0;8;0;0;8.5;5.1;7.65;7.4;0;10;4.75;8.1;8.5;7.7;6.5;8.8;8.77;7.5;8.8;8.8;10;8;8.4;9.25;7.15;8.1

10;0;9.13;0;0;8.1;6.2;0;0;0;8.5;7.8;9;0;8;0;9.48;7.9;8.3;5.6;0;0;0;7.9;0;0;0;0;0;0;0;0

10;10;9.53;8.3;0;7.8;7.4;10;0;9;9.34;0;9.3;8.7;0;8.6;0;0;0;0;0;0;0;0;0;0;0;0;0;10;0;0

8.5;7.2;10;7.9;9.3;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.4;8.6;8.8;9.3;7;7.15;4.55;7;4.95;4.15;6.8;7.7;7.61;7.7;5.8;5.8;6.5;3.58;8;5;6;7.7;5.755;7.5;5.2;2.7;10;5.8;5.2;9.56;6.39;7.2

8.6;7.4;8.8;7;4.55;7.4;4.6;7;2.83;6.5;8.5;6.4;8.6;6.3;0;5;7;8.7;9;4.4;0;5.05;5.04;6.9;5.5;0;8;7.1;5.7;9.25;7.7;8

7.9;5;7.5;8.4;5.05;7.5;5.3;8;0;7.5;5.3;6.7;8.9;6.8;0;2.72;7;6.5;6;6.2;0;0;5.03;0;6;9.5;0;0;0;7;0;0

7.5;5.4;9.72;5;2.05;3.4;0;9.5;0;0;0;0;8.4;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.5;9.3;9.97;8.1;9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;8.3;8.5;8.8;0;3.1;7;8;6;0;0;7.5;0;0;5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;2.5;0;0

0;0;6.06;7.1;0;0;0;9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

7.9;5.15;8.5;8.1;3.7;3.1;3.7;7.8;2;0.8;0;0;8.4;5.4;0;3.51;6.4;0;6.2;2.8;0;0;0;0;0;0;9;0;0;0;0;0

7;8.5;9.84;6;8.8;8;6;9.5;5.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8;3.75;7.6;9.4;3;5.5;3.95;7.5;0;5.6;8.2;4.12;9;6;0;5.18;9;0;8.8;1.6;0;0;8.28;7.1;0;0;9.5;0;0;7.5;8.5;7

9;5.35;7.4;8.2;4.55;8.8;5.9;8;7.5;6.2;8.3;8.65;8;8.3;7;9.4;9.5;7.4;8.8;9.7;6.6;0;0;0;9.5;8.8;0;0;0;0;0;0

0;0;6.72;0.5;0;0;1.9;8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

6;9.1;8.7;1.8;5.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.5;7.1;9.2;10;0;0;7.2;9.6;0;0;0;0;8.92;0;0;0;8.5;0;9;7.3;0;0;0;0;0;0;10;0;0;7.5;0;0

0;0;0;0;6;0;0;0;0;0;0;0;6.19;0;0;0;0;0;0;2.4;0;0;0;0;0;0;10;0;0;0;0;0

6;5.9;6.93;7.3;6.7;3.1;7.2;0;5.6;2.7;0;0;7.7;0;0;2;0;0;8.8;0;0;0;0;0;0;0;0;0;0;0;0;0

7.5;6.3;8.53;5.05;3.75;5.8;6.5;9;0.8;3.5;6.52;0;6;4;0;1.3;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.1;5;8.1;7.4;4.6;5.6;5.1;8;0;8.1;6.6;4.41;8.9;3.4;0;2.1;7.6;0;9.2;0;0;0;0;0;0;0;9;0;0;7.8;0;0

6.5;6.6;9.76;5;0;8.3;8;0;0;0;0;0;7.3;0;7.62;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.5;10;10;6.3;5.4;8.9;7.8;10;9;0;6.26;0;8.3;7.2;0;9.3;0;0;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.7;7.7;9.44;7.01;6.7;8.1;7.5;10;3.3;9.8;6.9;7.38;8.4;7.1;0;0;9.8;0;9;0;0;0;0;0;0;0;9;0;0;0;0;0

9.4;6.5;8.6;8.7;3.65;5;3.7;7.2;0;0;1.2;6.4;6.08;0;0;0;0;0;8.5;6.4;0;0;0;0;0;0;0;0;0;0;0;0

8.7;7.5;9;9;7.5;5.6;5;8.5;4.1;5.1;8.3;9.2;7.1;6.1;3.75;9.2;7.5;8;8;9.1;5.1;7.7;8.32;6.9;7.6;7.2;10;7.5;7;8.75;7.8;5.5

8.6;10;10;6;7.7;7.8;3.1;9.5;8.5;0;0;0;0;0;0;0;0;0;7.5;0;0;0;0;0;0;0;0;0;0;0;0;0

6;8.5;9.62;7.3;6.4;10;8.4;9.8;8.1;6.1;0;7.1;8.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.7;6.35;9.6;6.87;6;2.6;3.9;8;0;0;0;0;8.6;0;0;1.6;0;0;9.2;0;0;0;0;0;0;0;0;0;9.34;0;0

5.7;5.2;6;9.5;0;6.35;5.45;7;0;6;0;0;8.7;6.3;0;0;8;0;8.4;2.4;0;0;0;0;0;0;10;0;0;9.13;0;0

9.5;6.8;8.2;8.1;8;3.1;7;8;0;4.5;6.16;0;8.8;5.9;0;4.25;0;0;9.2;1.7;0;0;0;0;0;0;10;0;0;8.65;0;0

8;7;9.98;6;7.9;7;6.2;9.8;7.2;7;0;3.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;10;0;0;0;7.6;9;0;7;0;0;7.3;0;0;8.9;10;0;9.6;9;0;8.3;10;7.7;7.6;7.9;10;8;8.6;7.8;9.5;9.6

9;3.2;6.3;7;3.85;7.6;0;8;0;0;4.5;2.7;6.21;0;0;0.5;7;0;9.2;1.4;0;0;6.455;5.5;0;9.5;10;5.48;5.6;7;5.33;5.4

10;9.7;10;9.3;8.2;9.7;8.6;8.75;7.4;9.4;6.9;7;6;7.1;7.2;7.3;7;6.6;8.5;6.3;0;7.1;6.12;8.4;0;0;8.5;0;0;8.4;0;0

8.7;0;0;0;0;0;0;0;0;5.9;9.12;0;0;0;0;8.2;6.5;0;0;0;0;8.4;7.8;6.5;0;0;3;6.2;0;0;0

9.6;9.3;9.9;9.5;9.5;10;10;9.8;10;9.7;9.5;10;8.87;8.1;10;10;10;8.1;9.6;9.8;9.8;9;10;8.8;8.65;8.8;10;10;9.1;9.5;9.5;8.8

8.6;7.8;8.7;8.7;6.4;5.2;6.8;7;5.7;7.6;7.3;4.37;7.5;5;5.05;6.47;8.7;0;8.7;4.75;7;0;6.64;0;0;8.5;9;8.5;0;9;9.8;7.9

8.7;8.5;9.47;7.43;9.9;10;7.6;10;0;8.6;8.5;9.32;8.1;8.5;0;0;9.8;9.1;0;7.6;0;0;9.1;8.1;0;0;0;0;0;0;0;0

9.5;2.9;8.1;8.4;2.2;0;0.65;2.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;8.8;0;7;8.2;0;0;6.3

0;0;3.65;1.42;4.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;6;5.685;8.8;0;7.7;10;9.27;0;0;0;9.4;0;7.75;0;0;0;10;0;0;0;0;0;0;0;0;0;0;0;8.69;0;0

0;0;9;0;0;0;0;8.5;0;5.2;0;0;5.6;0;0;6.4;5.6;0;9.1;5.2;0;3.1;0;7.7;6.9;6.5;0;9;6.6;7;4.55;7.7

9.1;9;10;8.8;8.4;9.5;8.3;9;7.7;7.6;8.9;5;7.63;8;7.4;9;8.5;7.2;10;3.85;5.15;7.6;9;9.1;7.6;9.5;10;7.7;7.4;8.5;6.42;7.2

6;10;9.97;7.7;9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

7.5;6;9.65;7;7.6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

0;0;0;0;0;0;0;9.3;0;0;0;0;0;0;0;7.5;0;0;0;0;0;0;0;0;0;0;0;0;0;7.2;0;0

10;10;9.86;8.7;9;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.6;8.5;9.5;8.8;0;7.2;8;0;5;7;8.2;8.12;5.83;6.4;7.4;9.3;8;7;10;8.6;5;8.2;8.68;8;8.5;8.4;8.5;6.8;7.7;0;8.4;7.3

8.9;7.1;8.4;9.5;4.5;4.45;3.65;6;0;0.3;1.8;0;7;0;0;2.4;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

5.5;6.8;9.65;5.9;6;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

10;7.4;9.8;8.5;5.2;8.5;7.9;9.5;6.5;9.5;9.3;8.48;8.6;6.8;0;7.5;9.94;0;10;7.8;0;7.3;10;0;0;8.5;0;0;0;0;0;0

7.5;9.7;9.88;8.9;8.3;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8.9;6.6;9;9.2;6.5;8.3;7.7;7;7.6;0;7;7.11;8.3;7.1;8;8.18;9.4;7.8;6.6;6.4;6.2;8.3;0;7.9;0;0;9;0;0;0;0;0

8.7;8.6;8.9;8.5;9.2;9.6;8.6;7.6;7.9;7;10;9;7.83;8.6;6.2;9.5;8.5;7.4;8.8;7.6;7.6;7;8.92;9.2;8.35;8.2;10;8.3;7.4;9;9;0

5;7.5;9.2;8.3;9.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

6;0;9.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.4;6.5;9;8.2;5.2;4.5;0;0;0;0;0;0;8.6;0;0;0;6.7;0;0;5.3;0;0;0;0;0;0;0;0;0;0;0;0

8.9;5.1;8.3;9.5;7.8;7.1;6.1;7.2;0;8;9;5.3;7.42;6.8;0;7.2;9.5;7.7;0;7;0;0;9.04;8;0;6;8.5;7.1;8.3;9.5;7.7;0

8.9;7.7;8.5;7;9.5;7;5.6;7.9;6.3;6.5;9.2;9.3;7.03;6.9;6.7;10;8;6.9;9.3;6.4;6.6;7.9;8.92;8.7;8.25;8.8;10;8.2;8.9;7.8;8.8;8.9

9.2;0;10;0;0;0;6;7.25;0;0.3;5.1;0;7.1;9.4;2;5.9;9;0;8.5;5.7;0;0;0;6.9;7.7;7.6;10;7.5;0;7;7;7

10;4;9.88;3.1;0.3;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

7.4;4.3;7;6.9;3.9;6.25;5.55;7.75;1.45;8.2;7.72;7;8.8;6.5;0;5.05;9.51;0;8.6;0;0;0;7.96;0;0;0;0;0;0;0;0;0

8.6;8.2;8;8.7;8.1;6.3;8.5;8;8.4;10;7;8.4;8.6;0;8.5;8.1;7;8.2;8.1;4.4;0;7;9.25;7.4;7.8;0;0;0;0;0;0;0

8.5;10;9.49;5;7.5;8;6.2;9.5;7;0;0;2.9;7.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

9.3;6.5;9;8.8;6.8;5.8;5;7.8;4.8;4;8;6.9;7.19;7.6;5;7.5;7.5;8.5;10;5.5;2.95;7.4;8.02;7.9;7.6;7.3;9;6.13;8.8;7;8.6;7.7

7.9;7.7;8.1;8.5;9.3;6.3;5.4;7.4;8.4;7.1;8.5;8;6.03;8.2;9.3;8;8;8.1;10;5;7.1;5.6;7.41;7.3;6.5;8.8;10;7.5;5.4;7.6;8.2;7.3

8.5;6;10;7.8;2.7;7.1;5.4;9;0;0;0;2.2;8.1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

10;10;10;9.6;10;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

6.5;6.9;10;6.2;1.7;2.2;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

5.5;0;9;9.3;3.5;3.2;6.3;7.5;3.1;7.2;0;0;8.1;5.1;0;4.95;0;0;8.2;2.5;0;7.5;0;0;0;0;0;0;0;8;0;0

0;0;9;0;9;0;8.9;10;8.97;0;9.8;0;9.2;8;6.6;0;0;0;10;0;0;0;0;9.2;0;0;0;0;0;0;0;0

8.8;7.8;9.2;7.8;3.25;8.3;6;8.8;0;8.1;9.5;7.1;9;7.1;0;5.4;8.5;0;9.5;0;0;0;8.33;9.5;0;0;10;7.88;9.4;8.5;9.6;7.1

8;6.8;9.2;7.7;6.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

8;7.1;9.55;6.4;5.7;7.7;5.6;7;6;0;7.58;3.2;7.6;2.7;0;0;0;0;0;1.7;0;0;0;0;0;0;0;0;0;9.56;0;0

10;8.3;0;0;0;0;2.4;8;5.55;0;6.9;7.15;0;7.8;0;5.05;10;7.6;0;4.7;0;7.9;0;8.2;0;0;0;0;0;9;0;0

7.8;6.3;7.9;6;3.425;9;4.65;9;3.9;8.5;8.9;8.25;7.1;7;8.2;6.21;9.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

Caso 1

Arquivo alvo

7.8;6.3;7.9;6;3.425;9;4.65;9;3.9;8.5;8.9;8.25;7.1;7;8.2;6.21;9.5;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0

Arquivo aprovações

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

Caso 2

Arquivo alvo

9.2;8.4;8.9;8.6;7;8.6;5.4;7;6.8;8.8;9.6;6.7;7;9.6;7.5;7;9.5;8.5;9.5;7;7.2;8.4;9.23;8.1;0;0;9.5;0;0;8.5;0;0

Arquivo aprovações

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;0;0;1;0;0

Caso 3

Arquivo alvo

```
6.5;10;10;7;7;0;0;9.5;5.5;0;0;0;6.8;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
```

Arquivo aprovações

```
1;1;1;1;1;0;0;1;1;0;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
```

Caso 4

Arquivo alvo

```
9.2;8.4;8.9;8.6;7;8.6;5.4;7;6.8;8.8;9.6;6.7;7;9.6;7.5;7;9.5;8.5;9.5;7;7.2;8.4;9.23;8.1;0;0;9.5;0;0;
8.5;0;0
```

Arquivo aprovações

```
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;0;0;1;0;0
```

Caso 5

Arquivo alvo

```
7;9.5;9.7;5.3;6;8;7;7;9;0;9.5;8.3;7.5;7.4;8.15;5.6;0;0;8.5;7.2;0;0;0;0;0;0;0;0;0;0;0;0
```

Arquivo aprovações

```
1;1;1;1;1;1;1;1;1;0;1;1;1;1;1;1;0;0;1;1;0;0;0;0;0;0;0;0;0;0;0;0
```

Caso 6

Arquivo alvo

```
4.5;9;0;3.65;9.65;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
```

Arquivo aprovações

```
1;1;0;1;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
```

Caso 7

Arquivo alvo

```
10;10;9.83;9.2;7.3;10;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
```

Arquivo aprovações

```
1;1;1;1;1;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0
```

Caso 8

Arquivo alvo

```
9.5;7.1;9.2;10;0;0;7.2;9.6;0;0;0;0;8.92;0;0;0;8.5;0;9;7.3;0;0;0;0;0;0;10;0;0;7.5;0;0
```

Arquivo aprovações

1;1;1;1;0;0;1;1;0;0;0;0;1;0;0;0;1;0;1;1;0;0;0;0;0;0;1;0;0;1;0;0

Caso 9

Arquivo alvo

9.6;9.4;9.7;7;9.3;9.6;8.2;9;9.7;9.6;9.7;10;8;9;10;10;9.5;8.4;9.4;8.2;9.5;9.5;9.29;9.2;8.85;8.8;10;10;9.7;8.8;9.5;8.9

Arquivo aprovações

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

APÊNDICE C – Tabelas das diferenças entre Valores Reais e Valores Previstos dos testes

FSI	TP1	DPW	OC	MB	TP2	ILC	AEA	CALC1	ALG LINEAR	BD1	EDD1	SO	EDC	CALC2	PROB	AS
-1,96	-1,21	2,06	-2,41	-0,11	2,51	-3,89	0,60	0,00	-0,89	-1,34	-0,86	0,73	-0,46	0,00	-1,07	0,13
-1,22	0,79	-0,89	0,11	2,91	-0,61	1,49	0,12	-4,00	0,74	1,06	2,10	-0,51	-5,25	-0,25	3,19	0,27
1,04	-1,26	-0,96	2,37	-2,23	0,02	-0,59	0,40	-0,42	-0,11	1,16	-2,11	0,26	1,87	-0,89	-2,81	-0,62
0,45	0,81	-0,13	-0,93	-0,27	-0,43	-1,14	0,31	0,55	0,64	0,27	1,57	0,40	-0,97	-0,25	-0,53	-0,09
0,95	0,63	-0,09	-1,63	0,09	-0,20	-0,89	0,15	0,08	0,54	0,30	1,24	0,45	-0,56	-0,24	-0,95	-0,18
-1,93	-0,11	3,99	-2,24	-0,41	-0,68	0,75	-1,54	-0,23	0,14	-1,63	-1,10	-0,82	-2,84	0,00	1,49	0,45
0,04	0,18	-0,08	-0,09	-0,04	0,30	0,02	0,02	0,03	-0,13	0,93	-2,40	0,13	0,17	0,13	0,16	0,23
0,53	0,20	1,19	-0,44	-3,11	-0,48	3,05	-2,44	2,07	-0,11	0,39	0,73	-0,84	-1,47	-0,21	-0,05	-0,12
0,83	0,12	-0,88	-0,70	1,15	-0,14	-1,67	0,71	-0,86	0,53	0,21	1,37	0,57	-1,34	0,49	0,44	-0,04
4,68	-2,34	1,16	-3,90	-1,70	-0,62	-4,91	0,24	-2,44	-1,15	-1,85	5,92	1,18	-2,58	2,04	1,25	-0,70
3,47	-4,74	0,81	-0,11	-1,36	2,65	-5,24	0,66	-0,26	-0,90	0,15	-2,59	0,97	1,01	-0,68	0,60	-1,95
-1,96	0,00	3,52	-1,93	0,00	-0,67	-5,23	0,40	0,00	-0,93	-0,75	-0,54	1,03	-1,00	0,00	-4,09	0,21

EDD2	IHC	REDES 1	LFA	ESTAT	PCS	BD 2	AA	REDES 2	EMP	PM	PCS-SGBD	ADM FIN	PS	GPI
-0,40	-0,04	2,45	-3,30	0,00	-3,47	2,98	-5,03	-0,16	-0,13	0,89	0,18	0,21	1,14	1,10
2,61	0,18	1,07	5,34	-3,21	-2,15	-1,83	1,19	-2,51	0,02	-0,17	-1,87	-0,12	0,11	2,34
1,61	-0,16	0,93	2,60	-0,10	-1,68	-0,76	0,72	-0,67	-0,05	-1,67	1,44	-1,18	-0,53	2,40
-0,31	0,05	0,09	0,84	0,45	-1,33	0,06	0,45	-0,23	-0,03	-0,81	1,55	0,09	-1,21	1,37
-0,14	0,05	-0,10	1,29	0,60	-0,47	-0,91	0,11	-0,24	0,04	0,05	1,89	0,22	-1,71	1,09
-1,16	0,07	2,03	-5,31	0,23	3,98	1,28	1,02	-0,43	0,19	-1,89	0,45	0,73	-2,29	1,35
0,32	0,14	-3,09	-1,22	0,49	0,94	0,64	0,73	2,25	-0,07	-0,14	1,20	0,07	-2,58	1,02
-0,27	0,02	0,72	-2,00	0,33	0,44	-0,64	1,26	-0,30	0,16	-1,41	1,31	0,54	-0,78	0,93
-0,35	0,12	-0,97	0,01	0,15	0,06	-0,83	0,60	0,77	0,04	-0,86	1,61	0,01	-1,63	1,62
-2,98	0,25	-1,16	-1,83	0,58	5,17	-5,95	2,96	1,14	0,63	-0,87	1,56	0,62	-1,90	1,40
1,56	0,00	-0,25	0,99	-0,13	0,57	-2,02	2,34	0,15	0,12	0,08	1,94	0,06	-2,12	1,68
-0,41	0,02	1,58	-4,18	4,59	-3,61	5,84	-5,33	0,69	-0,37	-0,29	3,18	0,35	-2,16	2,89