# Concurso CareerBuilder Kaggle (2012): Uma proposta de recomendação colaborativa

Mauricio Noris Freire
Laboratório de Computação Natural e
Aprendizagem de Máquina LCoN
*Universidade Presbiteriana Mackenzie UPM*
São Paulo/Brasil
mauricionoris@gmail.com

***Resumo*** **—** Sistemas de recomendação são aplicados em diversos contextos como o de recrutamento digital. A CareerBuilder disponibilizou uma grande base de dados para um desafio no *Kaggle*. Usou-se uma abordagem de filtragem colaborativa baseada nas candidaturas em conjunto efetuadas para estabelecer interesses comuns entre os usuários. As similaridades foram geradas baseadas na distância de *Jaccard* e outras abordagens baseadas em conteúdo podem ser adicionadas à solução para refinar a recomendação.

***Palavras-Chave*** **—** *Sistema de Recomendação, Filtragem colaborativa, recrutamento digital, Kaggle, CareerBuilder*

***Abstract*** **—** Recommendation systems are applied in diverse contexts such as digital recruitment. CareerBuilder has released a large database for a challenge in Kaggle. A collaborative filtering approach was used based on the joint applications made to establish common interests among users. Similarities were generated based on the Jaccard distance and other content-based approaches can be added to the solution to refine the recommendation.

***Keywords***— *Recommender System, Collaborative Filtering, e-Recruitment, Kaggle, CareerBuilder*

## I. INTRODUÇÃO

Em 2012, o CareerBuilder.com patrocinou o desafio *Job Recommendation Engine Challenge* na plataforma Kaggle[1], que solicitou a previsão de quais vagas de emprego os seus usuários se candidatam baseados em suas aplicações anteriores, informações demográficas e histórico de trabalho. O seu objetivo era de aprimorar o algoritmo de recomendação de vagas aos candidatos em seu portal, que é uma parte central e um elemento-chave na experiência de seus usuários.

O interesse pelo desafio neste domínio específico se deve ao projeto de pesquisa científica que faço parte e é patrocinado pela FAPESP[2]. Neste projeto, busca-se desenvolver um motor (sistema) de recomendação que seja capaz de identificar os principais elementos e características das vagas e dos candidatos para que o melhor casamento (*matching*) possa ser feito entre eles. Assim, o objetivo deste trabalho é utilizar é validar abordagens como referencial a ser aplicado no banco de dados da empresa.

## II. REFERENCIAL TEÓRICO

### A. *Sistema de recomendação*

Sistemas de recomendação podem ser agrupados em três categorias distintas: a) baseadas em conteúdo (CB); b) filtragem colaborativa (CF); c) abordagens híbridas, que mesclam as duas anteriores. [1], [2]

Nas abordagens de filtragem colaborativa, existem situações que somente são possíveis capturar exemplos positivos no banco de dados. Esta situação é conhecida como Filtragem Colaborativa de Uma Classe (OCCF). Os cenários típicos desta situação são: a) marcadores sociais: marcadores sociais, onde cada usuário cria um conjunto de *bookmarks* contendo itens que podem ser consideradas como exemplos positivos. Porém, a ausência de outros itens não pode, por si só, ser considerado um exemplo negativo; b) histórico de cliques: os dados de cliques são amplamente utilizados em pesquisas personalizadas. Geralmente um triplo <u, q, p> indica um usuário u enviou uma consulta q e clicou uma página p. É comum que páginas que não foram clicados não são coletados. Semelhante ao exemplo de marcador, não se pode julgar se a página não é clicada por causa de irrelevância de seu conteúdo ou redundância. [3]

Diversas métricas de similaridade podem ser empregadas em sistemas de recomendação por filtragem colaborativa. Em [4], são apresentadas as principais opções e suas vantagens/desvantagens. São elas: a) (PCS) - *Pearson Correlation Similarity*; b) (CVS) - *Cosine Vector Similarity*, c) (MSD) - *Mean Squared Difference;* d) (FPC) - *Frequency-weighted Pearson Correlation;* e) (WPC) - *Weighted-Pearson Correlation;* f) (DSim) - *Discounted Similarity;* g) (SRC) - *Spearman Rank Correlation.* Em [5] é apresentada uma abordagem baseada na similaridade de *Jaccard* como uma forma de identificar os interesses comuns dos usuários, e assim, estabelecer seu grau de similaridade.

Apesar da ampla utilização destas similaridades, é sabido os sistemas de recomendação baseados em CF sempre sofrem com o problema de inicialização a frio, o *cold-start*, em que os usuários não têm classificações correlatas suficientes para efetuar previsão. Para abordar esses problemas, em [6] é proposta uma nova medida de similaridade inspirada em um fenômeno de ressonância física, denominada similaridade de ressonância (RES). Nela modela-se matematicamente a consistência dos comportamentos de classificação dos usuários, as distâncias entre as opiniões dos usuários e o fator de *Jaccard* com as classificações relacionadas e não relacionadas. RES é uma soma cumulativa do produto aritmético dessas três partes e é otimizada usando parâmetros de aprendizado de conjuntos de dados.

[1] https://www.kaggle.com/c/job-recommendation

[2] https://bv.fapesp.br/pt/auxilios/99765/um-sistema-de-recomendacao-personalizada-para-e-recruitment/

### B. E-Recruitment

Os principais requisitos para se adaptar um sistema de recomendação para *e-recruitment* são: 1) a correspondência de indivíduos para determinada vaga de trabalho depende das habilidades que os indivíduos devem ter; 2) recomendar pessoas é um processo bidirecional que precisa levar em conta as preferências não só do recrutador, mas também do candidato; 3) as recomendações devem basear-se nos atributos do candidato, bem como nos aspectos relacionais que determinam o ajuste entre a pessoa e os membros da equipe com quem a pessoa será colaborada; e 4) o indivíduo é considerado único, ou seja, não se pode escolher uma única pessoa várias vezes, como um filme ou um livro. [7]

Sistemas de recomendação sensíveis a determinado contexto (CARS) aprimoram os sistemas básicos, adaptando-as ao contexto específico do usuário. Os fatores contextuais que devem ser considerados por um sistema de recomendação estão relacionados ao momento, local e finalidade do usuário específico. [8]

Em [9] é proposta uma metodologia para recomendação de vagas que consiste em: a) seleção de características, que divide os atributos necessários para o *match* entre candidatos e vagas; b) categorização dos dados, com o objetivo de descobrir os critérios nos quais os candidatos, que pertencem a determinado grupo, decide por se candidatar a determinada vaga de emprego; e c) mineração de regras induzidas em uma árvore de decisão, filtram as vagas a serem recomendadas de acordo com regras obtidas nos dados; d) geração da recomendação, gera uma lista com as vagas com maior similaridade aos candidatos.

## III. PROBLEMA A SER RESOLVIDO

O objetivo é prever, para cada janela, a quais vagas os candidatos do grupo Teste se candidataram durante o período de teste da janela. Apesar de que os candidatos também podem ter se candidatado a trabalhos de outras janelas, o foco deve ser apenas nas candidaturas realizadas em suas próprias janelas.

A métrica de avaliação usada na competição foi o MAP@ 150 (*mean average precision*), que significa que é esperada uma lista com 150 vagas recomendadas para cada candidato. Em [10], explica-se esta métrica como a média das precisões em classificações individuais. Em outras palavras, a AP considera a precisão em cada resultado relevante na lista e a divide pela classificação do resultado; então, a precisão é calculada dividindo a soma das precisões descontadas pelo número total de resultados relevantes. Na maioria dos casos, o AP de muitas consultas é considerado e uma Precisão Média Média (MAP) é calculada por:

$$MAP = \frac{1}{|Q|}\left(\sum_{Qi}\frac{1}{|Ri|}\left(\sum_{j=1}^{n} rel(Dj)\frac{\sum_{k=1}^{j} rel(Dk)}{r_j}\right)\right) \qquad (1)$$

Onde:

- *Q* – Número de candidatos
- *R* – Itens relevantes (150)
- *D* – Documentos na posição do ranking *r* de *n* itens retornados
- *Rel* – Função de relevância (retorna 1 para os documentos relevantes)

Assim, pode-se concluir que: a) recomendar no máximo 150 itens para cada usuário e esta métrica beneficia o envio todas as recomendações encontradas, porque não se penaliza por suposições ruins; b) a ordem das recomendações é importante, por isso, é melhor enviar algumas recomendações com maior probabilidade primeiramente, seguidas por recomendações menos seguras.

Entretanto, como o gabarito de avaliação não está disponível, não será possível avaliar a qualidade das recomendações.

Em resumo, os dados são sobre candidatos, postagens de emprego e candidaturas realizadas. No total, as candidaturas abrangem 13 semanas. As bases estão divididas em 7 grupos, cada grupo representando uma janela de 13 dias.

Cada usuário é atribuído a uma janela com probabilidade proporcional ao número de aplicações realizadas por ele a trabalhos no período que compreende a janela. Por exemplo, se determinado usuário só aplicou para trabalhos da Janela 1, este será atribuído e ela com probabilidade de 100%. Um outro usuário, no entanto, aplicou para trabalhos na janela 1 e na janela 2 e, portanto, pode ter sido atribuído tanto para cada uma dessas janelas, com a probabilidade dividida entre elas.

É fornecida a lista de aplicações realizadas pelos candidatos na janela durante o período de treinamento. Os candidatos foram divididos em dois grupos: a) Teste ; e b) Treino. Os candidatos do segundo grupo são aqueles que fizeram 5 ou mais aplicações no período de teste de 4 dias e os candidatos do segundo são aqueles que não tiveram tantas aplicações no mesmo período.

O banco de dados está estruturado como arquivos com extensão TSV (*tab separated value*) e está estruturado da seguinte forma:

1. **jobs.tsv,** que contém informações sobre postagens de emprego. Cada linha deste arquivo descreve um posto de trabalho. Os usuários só podem se inscrever em um trabalho entre o *StartDate* e o *EndDate*. A Tabela 1 apresenta a estrutura deste arquivo.
2. **Users.tsv**, contém informações sobre os usuários. Cada linha deste arquivo descreve um usuário. A coluna *UserID* contém um número de ID exclusivo do usuário, a coluna *WindowID* contém de qual das 7 janelas o usuário está atribuído e a coluna Split informa se o usuário está no grupo *Train* ou no grupo *Test*. As colunas restantes contêm informações demográficas e profissionais sobre os usuários. A Tabela 2 apresenta a estrutura deste arquivo.
3. **apps.tsv** contém informações sobre as candidaturas feitas. Cada linha descreve uma candidatura. A Tabela 3 apresenta a estrutura deste arquivo.
4. **Users_history.tsv** contém informações sobre o histórico de trabalho de um usuário. Cada linha deste arquivo descreve um trabalho que um usuário mantinha. A Tabela 4 apresenta a estrutura deste arquivo.
5. **window_dates.tsv** contém informações sobre o tempo de cada janela. Cada linha corresponde a uma janela e tem a data e a hora em que o período de treinamento começa, que o período de treinamento termina e que o período de teste termina. A Tabela 5 apresenta a estrutura deste arquivo.

| Jobs.tsv | Dicionário de dados | | |
|---|---|---|---|
| | ***Tipo*** | ***Descrição*** | ***Estruturado (S/N)*** |
| Jobid | inteiro | ID vaga | S |
| WindowId | inteiro | ID janela | S |
| Title | texto | Título do cargo | N |
| Description | texto | Atribuições | N |
| Requirements | texto | Requisitos mínimos | N |
| City | texto | Localidade | S |
| State | texto | Estado | S |
| Country | texto | País | S |
| Zip5 | inteiro | Código postal | S |
| StartDate | data | Início do anúncio | S |
| EndDate | data | Fim do anúncio | S |

*Tabela 1 - Estrutura Jobs.tsv*

| Users.tsv | Dicionário de dados | | |
|---|---|---|---|
| | ***Tipo*** | ***Descrição*** | ***Estruturado (S/N)*** |
| UserID | inteiro | ID candidato | S |
| WindowID | Inteiro | ID janela | S |
| Split | texto | Treino/Teste | S |
| City | texto | Cidade | S |
| State | texto | Estado | S |
| Country | texto | País | S |
| ZipCode | inteiro | Código postal | S |
| DegreeType | texto | Últ. formação | S |
| Major | texto | Área formação | N |
| GraduationDate | data | Data graduação | S |
| WorkHistoryCount | inteiro | Qtd. trabalhos | S |
| TotalYearsExperience | inteiro | Experiência | S |
| CurrentlyEmployed | bool | Situação atual | S |
| ManagedOthers | bool | Gerenciamento | S |
| ManagedHowMany | inteiro | Tamanho do time | S |

*Tabela 2 - Estrutura Users.tsv*

| apps.tsv | Dicionário de dados | | |
|---|---|---|---|
| | ***Tipo*** | ***Descrição*** | ***Estruturado (S/N)*** |
| UserID | inteiro | ID candidato | S |
| WindowID | inteiro | ID janela | S |
| Split | texto | Treino/Teste | S |
| ApplicationDate | data | Data Candidatura | S |
| JobID | inteiro | ID Vaga | S |

*Tabela 3 - Estrutura apps.tsv*

| Users_history.tsv | Dicionário de dados | | |
|---|---|---|---|
| | ***Tipo*** | ***Descrição*** | ***Estruturado (S/N)*** |
| UserID | inteiro | ID candidato | S |
| WindowID | inteiro | ID janela | S |
| Split | texto | Treino/Teste | S |
| JobTitle | texto | Título vaga | N |
| Sequence | inteiro | Ordem decrescente | S |

*Tabela 4 - Estrutura Users_history.tsv*

| Window_dates.tsv | Dicionário de dados | | |
|---|---|---|---|
| | ***Tipo*** | ***Descrição*** | ***Estruturado (S/N)*** |
| Window | inteiro | ID janela | S |
| Train Start | data | Início do treino | S |
| Train End/Test Start | data | Início teste | S |
| Test End | data | Fim teste | S |

*Tabela 5 - Estrutura window_dates.tsv*

## IV. SOLUÇÃO PROPOSTA

### *A. Análise descritiva dos dados*

O arquivo *jobs.tsv* é composto por 1.054.348 vagas, sendo que 99,87% delas no território americano e 0,13% espalhadas em diferentes partes do mundo. As vagas internacionais serão desconsideradas no modelo de predição. A Figura 1 ilustra a distribuição de oportunidades de trabalhos postadas para cada estado americano e a tonalidade das cores representa esta densidade.

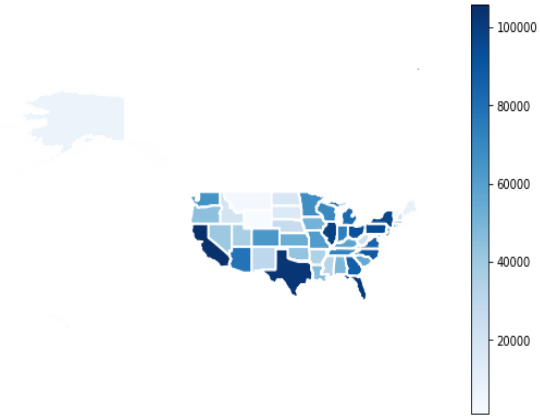


*Figura 1 - Distribuição das ofertas de trabalho*

O arquivo *users.tsv* é composto por 389.708 candidatos, sendo que 99,69% é composta por residentes nos Estados Unidos e 0,31% é composta por indivíduos residentes em outras localidades ao redor do mundo. Os indivíduos não residentes nos Estados Unidos serão desconsiderados no modelo de predição. A Figura 2 ilustra a distribuição de candidatos por estado americano e a tonalidade das cores representa sua densidade.


*Figura 2 - Distribuição de candidatos por estado*

Na comparação entre a distribuição de vagas com a distribuição de candidatos por estado americano, observa-se que em algumas regiões existe um desiquilibro significativo. A Figura 4 ilustra esta comparação, representando nos estados representados na tonalidade mais escura a escassez de mão-de-obra em relação ao conjunto de oportunidades e nos estados com a tonalidade mais clara um equilíbrio entre a quantidade de oferta de trabalho e a quantidade de candidatos. A probabilidade de sucesso no preenchimento das vagas é, em termos gerais, influenciada por este indicador.

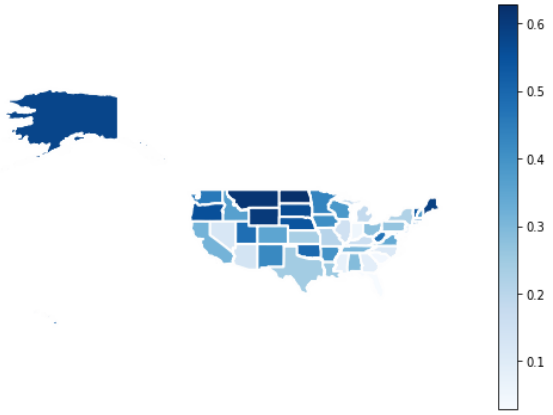


*Figura 4 - Vagas por Candidato*

A Figura 3 ilustra o conjunto de candidaturas realizadas em cada estado. A informação em azul representa as candidaturas realizadas por candidatos que residem no mesmo estado em que a oferta de trabalho se propõe. A informação em verde trata de candidaturas realizadas por profissionais que residem em um estado diferente daquele da oportunidade de trabalho. Portanto, questões demográficas representam uma grande influência na probabilidade de match candidato/vaga. A única exceção a esta regra, é a capital federal dos EUA, o item DC do gráfico que apresenta mais candidatos não residentes do que residentes.

A Figura 5 ilustra a distribuição de vagas em função do número de candidaturas. Observa-se que a maior parte das vagas tiveram até 50 candidaturas, e algumas poucas vagas tiveram até 200 candidaturas.

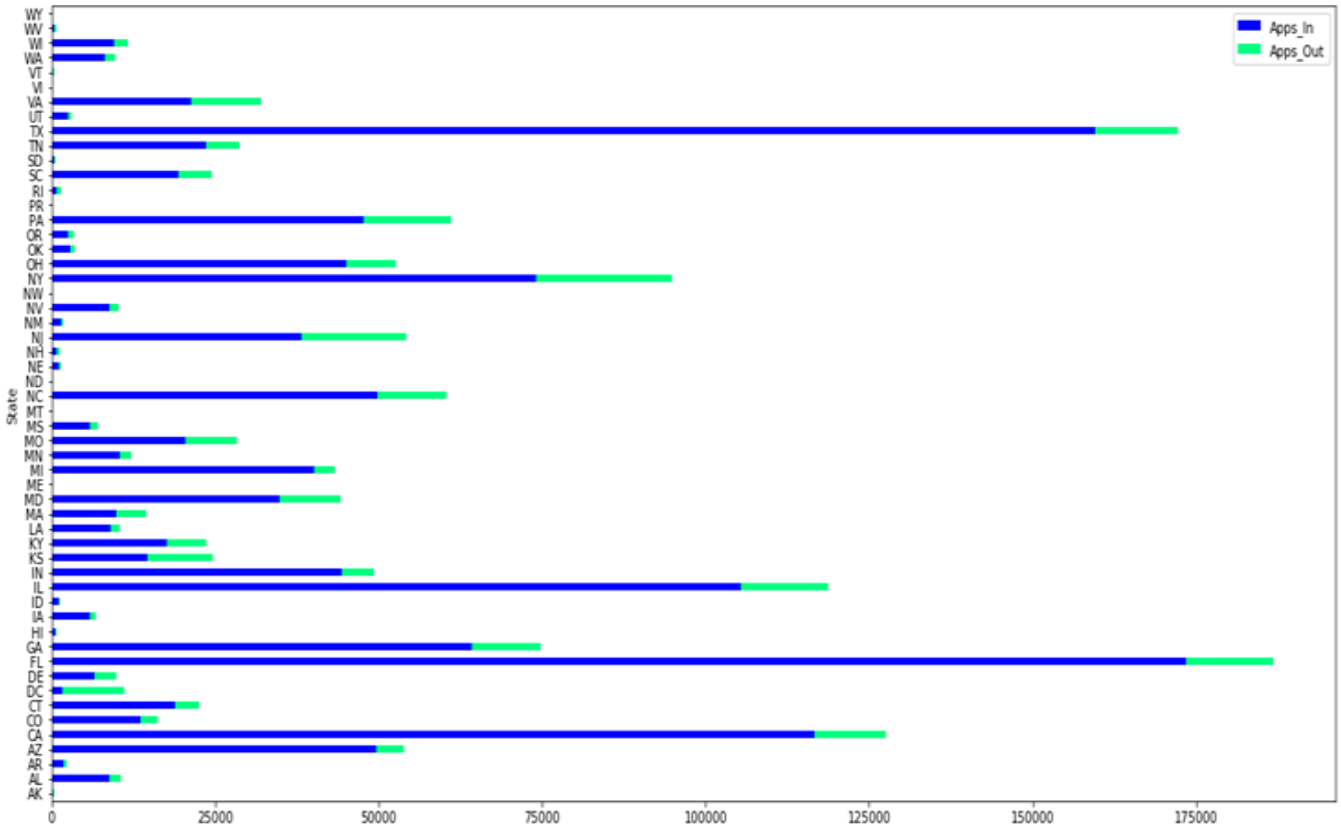


*Figura 3 - Candidaturas feitas no estado e fora dele*

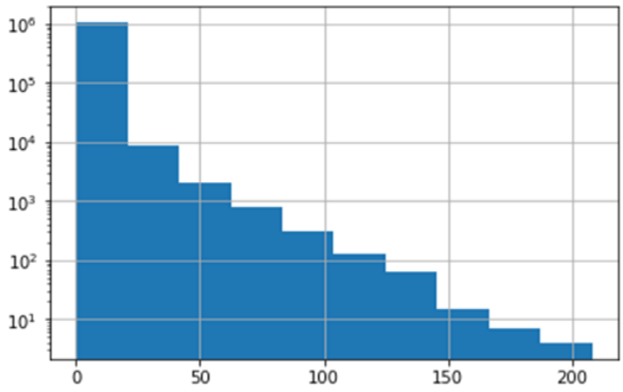


*Figura 5 - Vagas distribuídas pelo número de candidaturas*

O arquivo *users_history.tsv* contém 1.753.901 registros, referentes a 355.609 usuários. Uma vez que a base total de candidatos é de 389.708 candidatos, observa-se que aproximadamente 8,7% deles não possui qualquer informação sobre histórico profissional. Este subgrupo, composto por 34.099 pode ser subdividido em duas categorias: a) jovens profissionais em busca do primeiro emprego (42%) que corresponde a 14.158 usuários; e b) candidatos que não completaram corretamente o seu cadastro equivalente a (58%), ou 19.941 usuários. Embora exista esta lacuna nos dados, os candidatos do primeiro grupo irão permanecer na base de dados e poderão ser recomendados para vagas que não exijam tempo de experiência. Entretanto, os candidatos com perfil incompleto, serão ignorados uma vez que já possuem alguma experiência a qual não é possível identificar, e, portanto, inviável estabelecer sua preferência. A Figura 6 ilustra a composição do arquivo em relação às experiências profissionais dos candidatos. Nela é possível observar que a base de dados é composta em sua maioria por candidatos que tiveram quatro empregos ao longo da carreira, seguido por profissionais com três empregos e assim por diante. É possível observar também a presença de outliers, que relataram mais do que onze trabalhos ao longo da carreira.

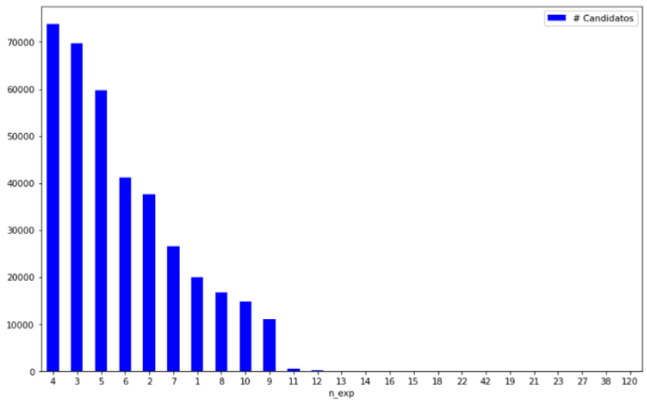


*Figura 6 – Distribuição de candidatos pela quantidade de experiências profissionais*

A Figura 7 ilustra o percentual de candidaturas por usuário. Observa-se que 65% usuários realizou entre uma e cinco candidaturas, 18% não realizou nenhuma ou realizou seis ou mais candidaturas. Percebe-se nesta distribuição uma diferença no sentido de urgência do conjunto de candidatos na conquista de uma nova vaga.

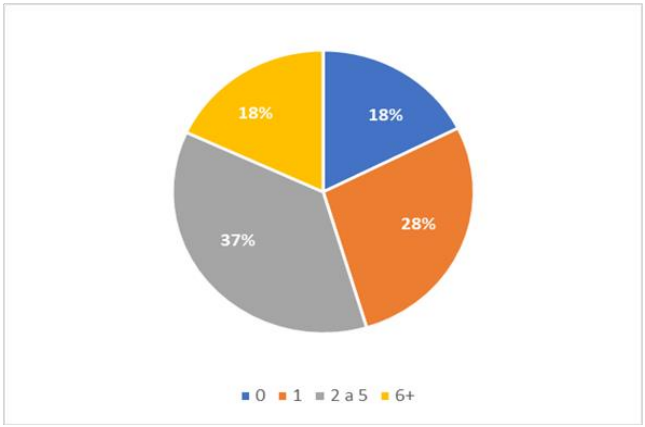


*Figura 7 - Distribuição de candidatos pelo número de candidaturas*

A Figura 8 ilustra o conteúdo do arquivo *window_dates.tsv*. As bases estão divididas em 7 grupos, cada grupo representando uma janela de 13 dias. Cada janela de 13 dias é dividida em duas partes: a) os primeiros 9 dias são o período de treinamento; e b) os últimos 4 dias são o período de teste. Dessa forma, cada anúncio de trabalho está atribuído a uma janela com probabilidade proporcional ao periodo em que estava no site nesta janela.

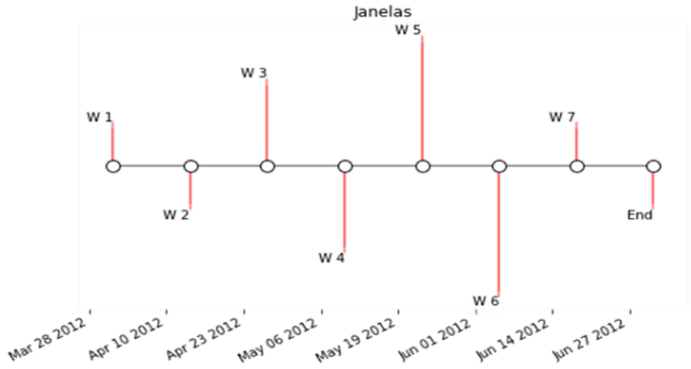


*Figura 8 - Linha do tempo*

A Tabela 6 apresenta a distribuição de candidatos entre as janelas. Pode-se observar que a distribuição entre as janelas e entre os períodos em cada janela são relativamente uniformes.

| **Janelas** | **Distribuição de candidatos** | | |
|---|---|---|---|
| | ***Total*** | ***Treino*** | ***Teste*** |
| 1 | 77.060 (19.77%) | 71.641 (93%) | 5.419 (7%) |
| 2 | 58.228 (14.94%) | 54.640 (93.8%) | 3.588 (6.2%) |
| 3 | 55.896 (14.34%) | 52.126 (93.3%) | 3.770 (6.7%) |
| 4 | 53.449 (13.72%) | 50.056 (93.7%) | 3.393 (6.3%) |
| 5 | 52.006 (13.34%) | 48.914 (94.1%) | 3.092 (5.9%) |
| 6 | 43.334 (11.12%) | 41.769 (96.4%) | 1.565 (3.6%) |
| 7 | 49.735 (12.76%) | 47.724 (96%) | 2.011 (4%) |
| Total | 389.708 | 366.870 (média 94,30%) | 22.838 (média 5.7%) |

*Tabela 6- Usuários em cada janela*

Os candidatos podem ser divididos em dois grupos: a) os que estão empregados atualmente; b) os que estão desempregados. A Tabela 7 apresenta a composição destes grupos. Observa-se que a quantidade média de candidaturas do grupo de candidatos desempregados é ligeiramente superior à média dos candidatos empregados. Porém, não é possível inferir que esta divisão influencia a probabilidade de uma candidatura. As Tabela 8 e 9 apresentam a subdivisão destes grupos, isolando os candidatos que fizeram candidaturas dos candidatos que não fizeram. Neste contexto, pode-se inferir que os candidatos que não realizaram quaisquer candidaturas no período não estão avaliando oportunidades e, portanto, serão excluídos do processo preditivo. Consequentemente, uma vez descartados os usuários que não fizeram quaisquer candidaturas, minimiza-se o problema de partida a frio.

| **Distribuição de candidatos** | | |
|---|---|---|
| ***Grupo*** | ***Total*** | ***Média*** |
| Empregados | 199.888 | 3.92 |
| Desempregados | 189.820 | 4.31 |

*Tabela 7 - Usuários em cada status*

| **Usuários por Status que realizaram candidaturas (Apps > 0)** | | | |
|---|---|---|---|
| ***Grupo*** | ***Total*** | ***% /Total*** | ***Média Candidaturas*** |
| Empregados | 163.102 | 81.60% | 4.81 |
| Desempregados | 189.820 | 83.61% | 5.17 |

*Tabela 8 - Usuários com candidaturas por status*

| **Usuários por Status que realizaram candidaturas (Apps = 0)** | | | |
|---|---|---|---|
| ***Grupo*** | ***Total*** | ***% /Total*** | ***Média Candidaturas*** |
| Empregados | 36.786 | 18.40% | 0 |
| Desempregados | 31.687 | 16.69% | 0 |

*Tabela 9 - Usuários sem candidaturas por status*

A Figura 9 ilustra a distribuição dos candidatos em função de sua formação acadêmica. Nela pode-se observar que pelo menos 50% dos usuários não possuem curso superior e que menos de 10% deles possuem pós-graduação. No modelo preditivo, será usado este parâmetro para estabelecer a probabilidade de uma vaga ser preenchida em função do nível de especialização requerido, por exemplo, um candidato com *Phd* pode estar disposto a aceitar desafios independentemente da localidade, desde que esteja à altura de suas capacidades e ambições. Por outro lado, usuários não tão especializados, tendem a permanecer próximo a suas localidades e sua disposição de se locomover pode até existir em função de outras variáveis, mas não necessariamente pelo cruzamento de informações. Portanto, o peso das questões demográficas na recomendação de vagas será inversamente proporcional à sua escolaridade.

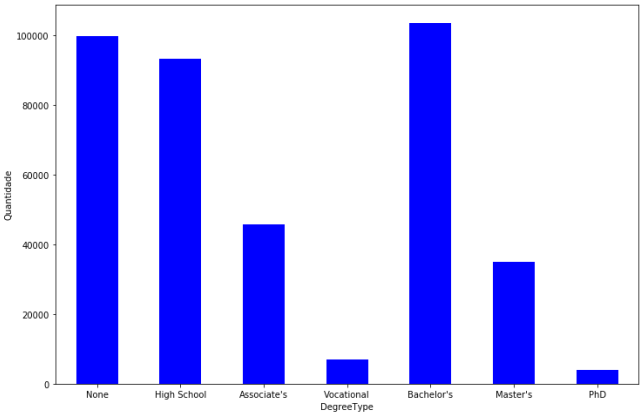


*Figura 9 - Distribuição dos candidatos pelo nível de escolaridade*

Um ponto a destacar nestes dados é que não há classificações explícitas. Em vez disso, há informações sobre em quais empregos o usuário se candidatou, independentemente do grau de afinidade. Por isso, este problema pode ser classificado como *oneclass collaborative filtering* (OCCF) porque aprende-se somente com a ação capturada, associada à ação de candidatura, e que pode ser considerado equivalente à uma classificação positiva do usuário para determinado item.

### *B. Preparação dos dados e pré-processamento*

A primeira atividade realizada no processo de preparação foi o cálculo da probabilidade de que uma vaga recebesse candidaturas em função do período em que está disponível em determinada janela. As Figuras 10 e 11 ilustram a distribuição de probabilidade das vagas, em todas as janelas dividida pelos períodos de Treinamento e Testes. O processo preditivo será influenciado pela probabilidade de que determinado usuário possa visualizar o anúncio de vaga.

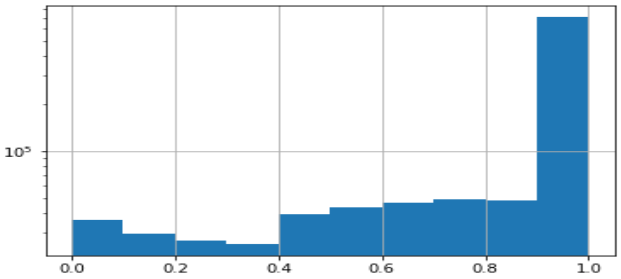


*Figura 10 - Distribuição de probabilidade visualização de uma vaga no período de treinamento*

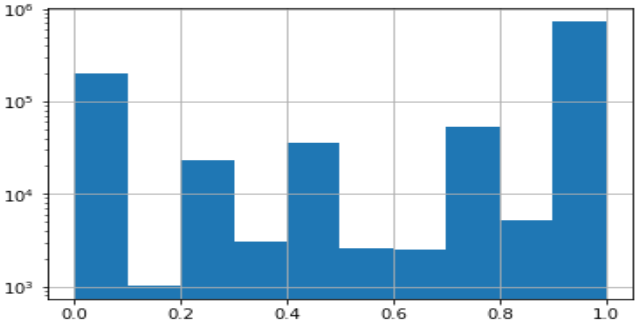


*Figura 11 - Distribuição de probabilidade visualização de uma vaga no período de teste*

As probabilidades foram calculadas em função do tempo de permanência na janela. Por exemplo, se uma vaga permaneceu ativa durante todo o período da janela esta terá 100% de probabilidade, se ficou somente pela metade do período, esta obterá apenas 50% e assim por diante. Arredondando as probabilidades para auferir a quantidade de candidaturas em função da probabilidade, exibe-se uma distribuição estável, em que o número de candidaturas não mostra claro viés em função desta probabilidade. As únicas exceções são: a) vagas que ficaram visíveis a até 10% do tempo da janela; b) vagas que ficaram disponíveis por todo o período da janela. A Figura 12 ilustra esta distribuição em escala logarítmica. As barras em azul representam as candidaturas efetuadas no período de treinamento das janelas e as barras em laranja representam as candidaturas no período de teste.

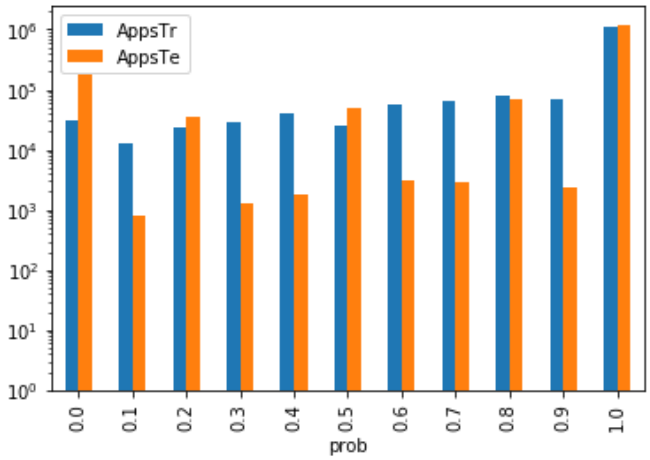


*Figura 12 - - Candidaturas feitas em função da probabilidade*

A formação acadêmica é uma característica que ocorre em certa ordem, onde a ordenação dos atributos é relevante e deve ser normatizada de modo a possibilitar o entendimento de seus dados. A Tabela 10 apresenta a normatização sugerida.

| **Users.tsv -> DegreeType** | | |
|---|---|---|
| ***Título*** | ***Descrição*** | ***Novo valor*** |
| *None* | Candidatos sem escolaridade | 0 |
| *High-school* | Candidatos com baixa escolaridade, que não possuem curso superior | 1 |
| *Associate's e Vocacional* | Representam candidatos que possuem cursos técnicos ou graduações de curta duração | 2 |
| *Bachelor's* | Candidatos que possuem curso superior | 3 |
| *Master's e Phd* | Candidatos que possuem pós-graduação | 4 |

*Tabela 9 - Normalização campo formação*

Entretanto, não existe um correlato estruturado no arquivo de vagas, sendo que o *match* entre formação dos candidatos, seu nível de experiência somente poderá ser comparado com o campo com a descrição de requisitos mínimos, que não é estruturado. Se neste campo for encontrada uma palavra que faça menção a algum destes Títulos, poder-se-á associá-los e daí filtrar os candidatos em função de sua formação, mas, caso contrário, o filtro não ocorrerá e atribui-se o valor padrão de zero, como se fosse irrelevante a formação do usuário, e daí o escopo de usuários sem formação acadêmica terão preferência nessas vagas.

A descrição do curso em que o usuário se formou (coluna *Major*) é um campo não estruturado e não foi encontrada um método eficiente de limpá-los. Fazendo uma análise do seu conteúdo, encontra-se 46.841 registros únicos, sendo que apenas 1.517 deles aparecem com uma frequência superior a 10 registros.

O título das ocupações encontradas no arquivo *users_history.tsv* é um campo não estruturado e seu conteúdo está demasiadamente disperso, gerando 657.155 registros únicos.

Tais características somente serão usadas em uma abordagem baseada em conteúdo, comparando com os outros campos não estruturados do arquivo de vagas, a saber: título, descrição e requisitos mínimos; em busca de alguma similaridade. Para este modelo, focado em filtragem colaborativa, estes atributos não serão avaliados.

### *C. Recomendação baseada em filtragem colaborativa (CF)*

A comparação entre os usuários será medida inspirada pela similaridade de *Jaccard*, que apura as vagas em que ocorre a intersecção, isto é, vagas que ambos se candidataram, sobre a união, que é o total de candidaturas de ambos. A Equação (2) apresenta a medida de similaridade para a parte de filtragem colaborativa da solução.

$$S(U_a, U_b) = \frac{U_a \cap U_b}{U_a \cup U_b} \qquad (2)$$

Onde:

- $U_a \cap U_b$= vagas que ambos se candidataram;
- $U_a \cup U_b =$ soma de todas as suas candidaturas

Considerando este índice de similaridade obtido entre os usuários, será possível recomendar vagas em que um dos dois não tenha se candidatado. Vislumbra-se com esta técnica colaborativa capturar contextualmente o interesse momentâneo comum dos candidatos.

As preferências dos usuários dependem do contexto e serão construídas gradualmente à medida que os usuários forem expostos a mais informações de domínio. [11]

Ora, se ambos realizaram a candidatura para uma mesma vaga, infere-se que possuem desejos semelhantes e, portanto, podem ser considerados similares naquele contexto, independentemente de possuírem capacidades e conhecimentos similares. Entretanto, não se está julgando neste caso, se um candidato tem maior afinidade com a vaga do que outro, mas, que ambos compartilham um interesse em comum. Pode-se comparar esta técnica à análise de visualização de produtos em um site de comércio eletrônico. Recomendações do tipo: "*clientes que viram este produto, também viram este"* funcionam na mesma lógica, daí a fórmula seria agrupar os usuários que navegaram em um item

em comum e efetuar a recomendação cruzada dos itens não simultâneos.

A competição exigia a recomendação dos candidatos realizados no período de *Teste* apenas. Portanto, para esta parte da solução, somente serão consideradas as aplicações realizadas por candidatos associados a este período em cada janela.

A Tabela 10 apresenta as candidaturas totais em cada período e por candidatos únicos. Nela também se observa o número de comparações entre os usuários realizadas durante o processamento.

| **Candidaturas realizadas (Período Teste)** | | |
|---|---|---|
| ***Janela*** | ***Dimensão*** | ***# comparações*** |
| *1* | Total: 49.749 candidaturas realizadas por 3138 candidatos | 4.921.953 |
| *2* | Total: 28.311 candidaturas realizadas por 2052 candidatos | 2.104.326 |
| *3* | Total: 27.618 candidaturas realizadas por 2190 candidatos | 2.396.955 |
| *4* | Total: 27.772 candidaturas realizadas por 2046 candidatos | 2.092.035 |
| *5* | Total: 27.718 candidaturas realizadas por 1941 candidatos | 1.882.770 |
| *6* | Total: 11.822 candidaturas realizadas por 878 candidatos | 385.003 |
| *7* | Total: 12.607 candidaturas realizadas por 957 candidatos | 457.446 |

*Tabela 10 – Candidaturas em cada janela*

A complexidade computacional para realizar o processamento é ilustrada pela Equação (3). Observa-se um crescimento numa escala pouco inferior à escala exponencial.

$$O(f(n)) = \frac{n \times (n-1)}{2} \qquad (3)$$

Onde:

- n = Quantidade usuários com candidatura

A Figura 13 ilustra a curva de crescimento das comparações. A complexidade é influenciada apenas pelo número de candidatos diferentes na janela, independentemente da quantidade de candidaturas realizadas.

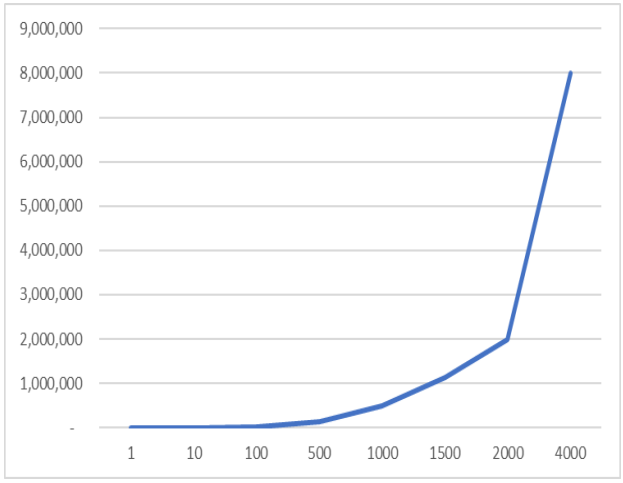


*Figura 13 - Crescimento do número de comparações*

## V. METODOLOGIA EXPERIMENTAL

### *A. Cálculo da similaridade*

O cálculo da similaridade entre os usuários foi executado pelo pseudocódigo apresentado abaixo.

```
Arquivo Saída:
     1.   W – código da janela (de 1 a 7)
     2.   Usr1 – código do usuário 1
     3.   Usr2 – código do usuário 2
     4.   Qt1 – # candidaturas usuário 1
     5.   Qt2 – # candidaturas usuárias 2
     6.   QtJ – # candidaturas em Conjunto
     7.   S – % similaridade

1.   Faça de 1 a 7:

1.1. Apps[w]  = Candidaturas da janela [w] do período de Teste

1.2. Users[w] = Agrupar por usuários únicos

1.3. Em cada item de Users[w] (1.2): 0 a n -1

     1.3.1.  Usr1 = Localiza o usuário[u1] em Users[w]
     1.3.2.  apps_u1 = candidaturas de Usr1 em Apps[w]

     1.3.3.  Em cada item de Users[w] (1.2): u1+1 a n

       1.3.3.1.  Usr2 = Localiza o usuário[u2] em Users[w]
       1.3.3.2.  apps_u2 = candidaturas de Usr2 em Apps[w]
       1.3.3.3.  inner = Candidaturas em comum (Usr1 e Usr2)
       1.3.3.4.  union = Total Candidaturas (Usr1 e Usr2)
       1.3.3.5.  S = inner / union
       1.3.3.6.  Se S > 0, adiciona no arquivo de saída

1.4.  Salvar arquivo saída da janela_[w].tsv
```

Uma vez identificados os candidatos similares, estes são ordenados de forma decrescente pelo índice calculado. A fim de evitar ruídos na recomendação e torná-la mais homogênea, uma vez que nem todos os usuários possuem a mesma quantidade de usuários similares, utiliza-se apenas os primeiros 5 usuários mais parecidos.

A Tabela 11 apresenta a quantidade de candidatos únicos onde se identificou pelo menos um usuário similar e possibilitou a geração de recomendações. Duas situações explicam a quantidade de usuários que não têm similaridade com qualquer outro, são elas: a) candidato não efetuou qualquer candidatura (*coldstart*); e b) candidaturas realizadas em vagas as quais ninguém mais se candidatou e, portanto, não há como calcular a similaridade.

| Candidatos | | |
|---|---|---|
| ***Qtd. Usuários Similares*** | ***Dimensão*** | ***% Total*** |
| *1* | 1917 | 20% |
| *2* | 1325 | 13.8% |
| *3* | 906 | 9.5% |
| *4* | 757 | 7.9% |
| *5+* | 4681 | 48.8% |
| ***Total*** | **9.586** | **Observação**: representam 42% dos 22.838 usuários analisados |

*Tabela 11 - Distribuição pelo número de candidatos similares*

Abaixo está o pseudocódigo usado para filtrar os arquivos para obter apenas os cinco maiores usuários similares.

```
Arquivo Saída:
     1.   Usr_1      - Usuário 1
     2.   Usr_2      - Usuário 2
     3.   Similarity - Valor similaridade

1.   Faça de 1 a 7:
1.1. Abra o arquivo janela[w]
1.2. Filtre-o: considerar similaridades entre 0.05 e 0.99
1.3. S_all += concatenar todos em uma lista única

2.   S_All_Clone: Clonar o S_all invertendo Usr_1 e Usr2
3.   S_F = Concatenar S_all com seu clone
4.   Agrupar por Usr_1 e capturar apenas os 5 maiores similares
5.   Salvar arquivo saída da top5.tsv
```

## B. *Gerando recomendações*

As recomendações serão geradas comparando as candidaturas não comuns entre dois usuários similares. A Figura 14 ilustra o processo de recomendação proposto, onde as candidaturas em comum são representadas no centro da figura e a direção das setas representam as vagas recomendadas a cada usuário.

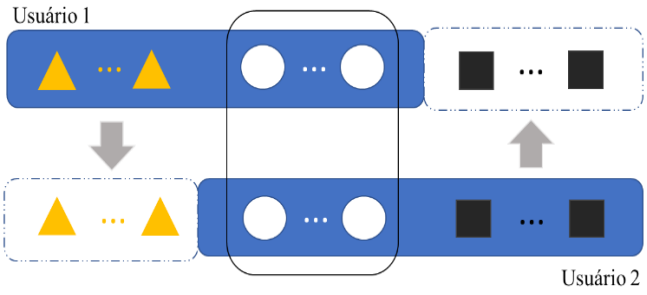


*Figura 14 - Recomendações*

O cálculo da similaridade entre os usuários foi executado pelo pseudocódigo apresentado abaixo.

```
Arquivo Saída:
     1.   W        - Código da janela (de 1 a 7)
     2.   Usr_To   - Usuário com a vaga recomendada
     3.   Usr_From - Usuário que partiu a recomendação
     4.   JobID    - Vaga Recomendada

1.   Faça de 1 a 7:
1.1. Apps[w]  = Candidaturas da janela [w] do período de Teste
1.2. Top5 = Lista de usuários com as 5 maiores similaridades
1.3. Em cada item de Top5Ranking (1.2): 0 a n

     1.3.1.  Usr1 = Coluna User1 em no Top5
     1.3.2.  Usr2 = Coluna User2 em no Top5
     1.3.3.  J1   = candidaturas de Usr1 em Apps[w]
     1.3.4.  J2   = candidaturas de Usr2 em Apps[w]
     1.3.5.  JTR1 = candidaturas de J2 (não em comum com J1)
     1.3.6.  JTR2 = candidaturas de J1 (não em comum com J2)
     1.3.7.  Adiciona JTR1 no arquivo saída: [J1,J2, JTR1]
     1.3.8.  Adiciona JTR2 no arquivo saída: [J2,J1, JTR2]

1.4. Salvar arquivo saída da recom_[w].tsv
```

# VI. RESULTADOS E DISCUSSÃO

## A. *Similaridades*

O resultado o cálculo de similaridades foi de 101.444 pares de usuários semelhantes. A Tabela 12 apresenta a distribuição destas similaridades por janela. Os que possuem o índice de similaridade igual a 100% não acrescentam qualquer informação útil ao modelo e, portanto, são descartados.

| Similaridades | | |
|---|---|---|
| ***Janela*** | ***Quantidade*** | ***Média (candidatos similares / candidato)*** |
| *1* | 31.244 | 9.95 |
| *2* | 14.940 | 7.28 |
| *3* | 17.400 | 7.94 |
| *4* | 14.507 | 7.09 |
| *5* | 15.491 | 7.98 |
| *6* | 3.600 | 4.1 |
| *7* | 4262 | 4.45 |

*Tabela 12 - Similaridades encontradas*

Avaliando individualmente as similaridades calculadas, observa-se que, em sua grande maioria, trata-se de um valor demasiadamente baixo, e por isso, é necessário limitar o entendimento de similaridade a um certo limiar. Adotou-se 5% como limiar mínimo para validar a associação entre dois usuários. A Tabela 13 apresenta a lista de similaridades descartando os valores abaixo do limiar escolhido.

| Similaridades ( > 5%) | | |
|---|---|---|
| ***Janela*** | ***Quantidade*** | ***Média (candidatos similares) / candidato*** |
| *1* | 9.722 | 3.09 |
| *2* | 5.590 | 2.72 |
| *3* | 6.939 | 3.1 |
| *4* | 5.514 | 2.69 |
| *5* | 5.232 | 2.69 |
| *6* | 1.406 | 1.6 |
| *7* | 1.667 | 1.74 |

*Tabela 13 - Similaridades acima do limiar*

### B. *Recomendações*

Foram geradas 747.863 recomendações exclusivas. Algumas vagas foram recomendadas a determinado usuário por mais do que um usuário semelhante. A Figura 15 ilustra esta situação, exibindo a distribuição de recomendações em função do número de vezes que uma vaga foi recomendada a certo candidato. Nela observa-se que a grande maioria das recomendações foi feita por apenas um único candidato similar, chegando a uma situação máxima em que doze candidatos recomendaram a mesma vaga a determinado candidato. Este indicador influencia a ordem de importância das recomendações.

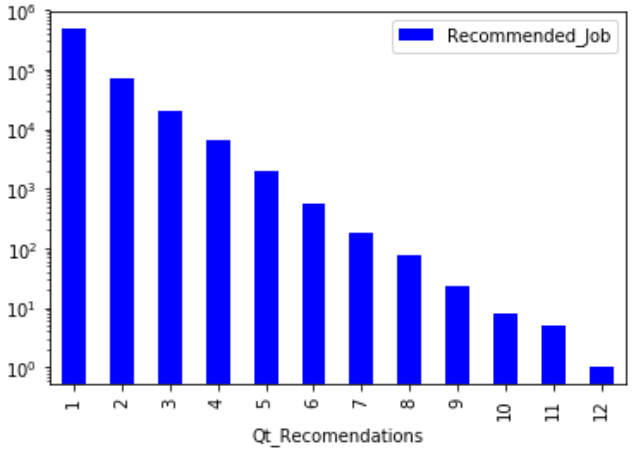


*Figura 15 - Distribuição de recomendações geradas*

A Figura 16 ilustra grupos de vagas recomendadas um certo grupo de candidatos. A maior parte das vagas foram recomendadas entre 1 e 20 candidatos e a menor parte recomendada até a 160 candidatos. Diferentemente do caso anterior, ilustrado na Figura 15, onde a recomendação simultânea reforça a probabilidade de pertinência da recomendação, aqui neste indicador, quanto maior a quantidade de usuários em que se é recomenda uma vaga, menor é a probabilidade de pertinência de tal recomendação.

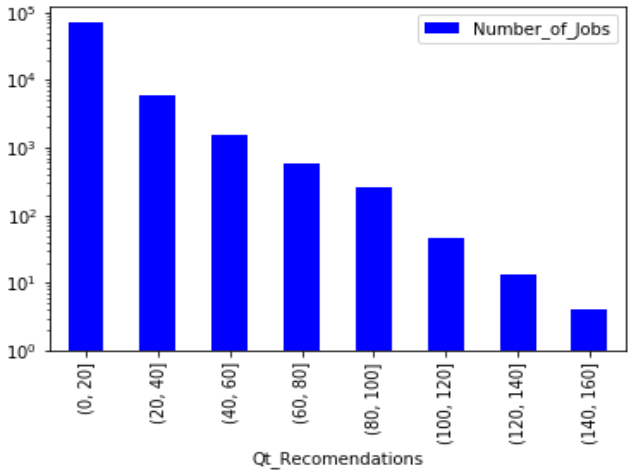


*Figura 16 - Quantidade de recomendações por vaga*

Portanto, a lista final de vagas recomendadas será refinada e as vagas ordenadas em função destes indicadores, promovendo vagas indicadas por mais do que um usuário similar e penalizando as vagas que forem recomendadas a um maior número de pessoas.

### C. *Limitações do modelo*

Observando-se a quantidade de candidaturas realizadas pelos usuários de teste, porque é neste subconjunto de usuários que é solicitado a recomendação, apenas cerca de metade deles fez pelo menos uma candidatura. Para a outra metade, sem dados presentes e, portanto, sem filtragem colaborativa.

A dispersão de dados também limita a capacidade de uso deste modelo exclusivamente. Os dados consistem em sete "janelas". Pode-se tratá-las separadamente, porque cada usuário e cada vaga é atribuído exatamente a uma janela sem existir sobreposições. Usando a Janela 2 como exemplo, observa-se que existem cerca de 50 mil usuários únicos e 50 mil trabalhos exclusivos e apenas 200 mil candidaturas. Isso significa que os dados são muito escassos: sua densidade é de aproximadamente 0,01%. Na comparação com o desafio Netflix[3], que existiam mais usuários, mas também muito mais avaliações (100M), então a densidade foi de cerca de 1%, 100 vezes maior do que neste desafio.

Portanto, para ser completa, a abordagem precisa levar em consideração características dos candidatos e não só suas ações. A solução completa deverá combinar os dados disponíveis na filtragem colaborativa (CF) e baseada em conteúdo (CB) quando não houver informação, constituindo uma solução híbrida.

## VII. TRABALHOS FUTUROS

O método apresentado aqui indica uma similaridade contextual, e, portanto, serve como um bom indicador de recomendação. Entretanto, não é uma solução suficientemente abrangente para todos os casos. Faz-se necessário adicionar outras abordagens, baseadas em conteúdo, que possam validar a recomendação colaborativa, descartando usuários não qualificados e/ou reordenando a recomendação de uma vaga em detrimento a outra.

Abaixo apresenta-se um conjunto de desdobramentos possíveis de serem adicionados à solução.

### A. *Recomendação baseada em conteúdo (CB)*

Dada a característica da base de dados, vislumbra-se a possibilidade três abordagens que podem ser usadas em conjunto para encontrar possíveis *matches*. São elas:

1. Título da vaga: pela comparação entre o título da vaga aberta com o título do histórico profissional dos candidatos.
2. Tratamento do campo formação (*Users.tsv*) e comparação com o campo não estruturado de requisitos mínimos (*Jobs.tsv*). Visa-se encontrar descrições de atributos de formação mínimo para agrupar as vagas e os candidatos em função do nível de escolaridade exigido. Esta solução

[3] https://www.netflixprize.com/

pode funcionar bem para vagas que exigem alta especialização.

3. Similaridade entre a vagas. Analisando atributos demográficos conjuntamente com a descrição das vagas poderá ser estabelecido um conjunto de vagas parecidas, e assim recomendar aos candidatos para uma vaga, outras possibilidades de recomendação a partir dela.

### *B. Unindo as soluções - Recomendação Híbrida*

Uma vez definido o conjunto de recomendadores, cada um estabelecendo um certo grau de similaridade, pela análise um certo aspecto das vagas e/ou candidatos, será possível concatenar esta informação, gerando uma lista resultante que leva em consideração diversos aspectos simultaneamente. Ora, parece intuitivo que uma vaga pode ser interessante sob um aspecto e desinteressante por outro, a informação fruto de um modelo frente ao outro poderá gerar um conjunto de vagas recomendadas que não sejam óbvias.

O peso dado a cada um dos recomendadores pode variar para cada usuário e a observância do uso e pertinência da recomendação, que pode ser medida pelo efetivo interesse despertado ao item recomendado, pode ser usada para calibrar a distribuição de pesos entre eles.

### *C. Aplicando estes conceitos na base de dados da empresa Reachr*

Independente da abordagem usada na construção do algoritmo de recomendação, faz-se necessário mediar sua pertinência. Para tal, pode-se perguntar ao usuário que recebe a recomendação, exigindo e confiando em um *feedback* explícito, para classificar determinado item recomendado em uma escala qualquer, pode não capturar completamente a opinião dos usuários sobre certo item, além de possuir um viés cognitivo singular para cada usuário. Ora, o que garante que um usuário interpreta da mesma forma a escala de classificação que outro? Então, como compará-los? Sendo assim, existe a possibilidade de coletar *feedbacks* implícitos dos usuários, observando seu comportamento em relação ao conjunto recomendado. A principal vantagem deste método é que não requer uma ação do usuário. Entretanto, tal modo de coleta de informações aumenta a complexidade na disponibilização de funcionalidades. [12], [13]

Outro ponto a se observar é o aspecto tempestivo de se fazer a recomendação apropriada. Em situações onde o processo de tomada de decisão depende do tempo de permanência (ou seja, o intervalo de tempo) entre decisões sucessivas, como em um cenário de transição de trabalho. Fazer recomendações no momento certo ajuda o sistema a obter maior utilidade. Utilidade é definida como a satisfação ou valor que um usuário obtém usando o sistema. Para motivar a discussão, considere a seguinte questão: quando o sistema de recomendação deve recomendar uma posição de engenheiro sênior de software para os engenheiros de software? É provável que o sistema atinja uma utilidade positiva quando um engenheiro de software que trabalha por dois anos recebe essa recomendação. No entanto, o sistema pode atingir uma utilidade negativa quando um engenheiro de software que trabalha por 2 meses ou 5+ anos recebe essa recomendação.[14]

Uma oportunidade para interpretar o momento de desenvolvimento profissional de um usuário seria modelar a similaridade entre os seus históricos profissionais. Esta medida pode ser usada em diversas aplicações como o planejamento de crescimento profissional, tanto para estudantes como para profissionais, além de permitir o entendimento dos caminhos, os marcos na carreira e uma atualizada avaliação da situação profissional do usuário. [15]

Sistemas de recomendação são geralmente avaliados por métricas que caracterizam as vantagens do sistema de um modo geral, mas pouco se presta atenção à possibilidade de recomendações ruins e não pertinentes aos usuários. Gerenciar a potencial perda de utilidade para cada usuário é crucial. Um exemplo ilustrativo deste problema é a recomendação para uma posição iniciante em determinado campo de trabalho para um profissional com ampla experiência. Em [16] é proposto um modelo baseado na análise do comportamento histórico dos usuários. Um entendimento destes interesses, personalizado para cada usuário, estabelece um limiar único em que se considera a utilidade mínima para cada um e só se faz a recomendação se esta condição é atingida. Para tal, o comportamento dos usuários é classificado em categorias, com base nas quais um algoritmo de otimização estima o custo/benefício da recomendação usando um algoritmo *bayesiano* de ordem parcial.

As abordagens existentes no momento apresentam isoladamente um certo benefício e podem satisfazer parcialmente a necessidade de aproximar candidatos às vagas. Porém, cada uma delas peca em certo aspecto e a não existe como satisfazer plenamente os requisitos com uma abordagem ou outra. Portanto, ao invés de um modelo único, sugere-se um conjunto motores de recomendação, atuando paralelamente, cada um estabelecendo um *ranking* de itens a recomendar. Este conjunto de *rankings* podem, pela presença de certo item em mais do que uma lista, corroborar a pertinência de sua recomendação. Este conjunto de algoritmos pode ser personalizado para cada usuário, onde atribui-se um peso específico para cada um deles, e assim, a recomendação entregue se adapte a cada preferência. Além disso, é crucial capturar o comportamento dos usuários durante a interação com o sistema, fazendo com que o sistema receba estes *feedbacks* implícitos e se ajuste em função deles.

Para cada item recomendado, deve-se identificar sua origem, ou seja, qual foi o algoritmo de recomendação que o recomendou. A análise de tais algoritmos se dará pela aceitação e pertinência dos itens recomendados por ele e permitirá a identificação de pontos de melhoria em desenvolvimentos futuros.

## REFERÊNCIAS


[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013.

[2] F. Ricci, L. Rokach, and B. Shapira, "Recommender Systems Handbook," in *Springer*, 2011, p. 845.

[3] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-class collaborative filtering," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2008, pp. 502–511.

[4] L. Al Hassanieh, C. A. Jaoudeh, J. B. Abdo, and J. Demerjian, “Similarity measures for collaborative filtering recommender systems,” *2018 IEEE Middle East North Africa Commun. Conf. MENACOMM 2018*, pp. 1–5, 2018.

[5] M. Ayub, M. A. Ghazanfar, M. Maqsood, and A. Saleem, “A Jaccard base similarity measure to improve performance of CF based recommender systems,” *Int. Conf. Inf. Netw.*, vol. 2018–Janua, no. January, pp. 1–6, 2018.

[6] Z. Tan and L. He, “An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle,” *IEEE Access*, vol. 5, pp. 27211–27228, 2017.

[7] S. T. Al-Otaibi, “A survey of job recommender systems,” *Int. J. Phys. Sci.*, vol. 7, no. 29, pp. 5127–5142, 2012.

[8] M. Eirinaki, J. Gao, I. Varlamis, and K. Tserpes, “Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions,” *Futur. Gener. Comput. Syst.*, vol. 78, pp. 413–418, Jan. 2018.

[9] A. Gupta and D. Garg, “Applying data mining techniques in job recommender system for considering candidate job preferences,” in *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, 2014, pp. 1458–1465.

[10] J. Griesbaum, T. Mandl, and C. Womser-Hacker, “An evaluation of popular relevance metrics,” 2011.

[11] J. W. Payne, J. R. Bettman, and D. A. Schkade, “Measuring Constructed Preferences: Towards a Building Code,” Kluwer Academic Publishers, 1999.

[12] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based Recommender Systems: State of the Art and Trends,” in *Recommender Systems Handbook*, 2011, pp. 73–105.

[13] L. Lerche, “Using Implicit Feedback for Recommender Systems: Characteristics, Applications, and Challenges,” 2016.

[14] J. Wang, Y. Zhang, C. Posse, and A. Bhasin, “Is it time for a career switch?,” 2016, pp. 1377–1388.

[15] Y. Xu, Z. Li, A. Gupta, A. Bugdayci, and A. Bhasin, *Modeling Professional Similarity by mining Professional Career Trajectories **. 2014.

[16] H. Gui, H. Liu, X. Meng, A. Bhasin, and J. Han, “Downside management in recommender systems,” *Proc. 2016 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2016*, pp. 394–401, 2016.