Mineração de dados sobre gênero na Wikipedia em Espanhol

Edinadja Macedo[1], Leilane Cruz[2]

[1] Centro de Informática da Universidade Federal de Pernambuco

[2]Centro de Informática da Universidade Federal de Pernambuco

{emm3, lccl2}@cin.ufpe.br

Resumo. *A Wikipepia é plataforma de referência na colaboração entre usuários na internet e seu uso resgata o espírito de compartilhamento inicialmente pensado quando da concepção da rede. Contudo, disparidades sociais se reproduzem entre seus editores: há uma diferença considerável entre a quantidade e a participação de editores homens e mulheres. Nesta análise, utilizamos a metodologia CRISP-DM para explorar o dataset coletado por Minguillón et al (2021) e responder à pergunta: "qual o provável gênero, dentre os presentes no dataset, dos usuários rotulados como desconhecidos?"*

Palavras-chave: Wikipedia, gênero, CRISP-DM.

1. Entendimento do domínio

1.1 Introdução: o problema da distribução de gênero na Wikipedia

A necessidade de organização e sistematização do conhecimento está longe de ser uma questão recente. Ainda no século XVIII, Denis Diderot e Jean le Rond D'Alembert iniciaram o movimento de criação da *Encyclopédie, Ou Dictionnaire Raisonné Des Sciences, Des Arts Et Des Métiers*, obra com 35 volumes escrita com a pretensão de catalogar todo o conhecimento humano (ENCICLOPÉDIA BRITÂNICA, 2022). Este esforço de pensadores iluministas, à época uma necessidade política de sistematizar o conhecimento secular, reeditado contemporaneamente por empresas como a Google[1], teve sua estrutura amplificada pelo uso aberto das plataformas de wiki, especificamente no projeto Wikipedia, mantido pela Wikimedia Foundation.

Com a proposta de democratizar o conhecimento e tornar a construção de artigos

1 Definição da missão da Google no site da empresa: "Our mission is to organize the world's information and make it universally accessible and useful." Disponível em: https://about.google/intl/ALL_us/ Acesso em: 21.05.2022

um processo colaborativo – bem mais amplo que a escrita restrita a especialistas da enciclopédia original e das subsequentes, o projeto possibilita o cadastro de usuários e editores, os quais debatem, de forma aberta e em comunidades, a inclusão de conhecimento aceito sobre determinado tópico (WIKIPEDIA, 2022). Um dos projetos mais citados como referência de concretização dos ideais de uso da internet[2], a Wikipedia conta com bilhões de visualizações por mês (WIKIMEDIA FOUNDATION, 2022) e já ultrapassou as iniciais contestações de credibilidade de seus artigos, tornando-se ponto de partida recorrente na realização de pesquisas.

Dado esse histórico, houve certo burburinho entre acadêmicos e comunidade interessada quando pesquisa da United Nations University - UNU-MERIT apontou disparidades no que Graham, Strauman e Hogan (2015) denominam "geografias da participação."[3] O estudo, que abrange marcadores demográficos como Idade, Educação, Status de Relacionamento e Gênero, indica que menos de 13% dos editores da plataforma são do gênero feminino (GLOTT; GHOSH, 2010). Esta notável desproporção participativa, a qual indica uma diferença objetiva entre a pretensão de acesso democrático e as práticas de uso da plataforma, inspirou uma série de trabalhos em diversas áreas, os quais exploraram desde disparidades gerais à questão específica do abismo entre colaboradores de diferentes gêneros.[4]

Em que pese trabalhos como o de Hill e Shaw (2013) terem revisitado a pesquisa inicial e observado possível viés na amosta, a repercussão do debate gerado pela estatística publicada inicialmente se manteve. A exemplo, o dataset escolhido para este exercício, explorado originalmente em artigo de 2021, investiga a diferença de participação de mulheres e homens na edição de artigos de língua espanhola na Wikipedia. O trabalho de Minguillón et al (2021) faz uma análise estatística da intensidade das contribuições por gênero e explora aprofundadamente as métricas da colaboração nestes artigos.

2 Propagados, por exemplo, por um de seus idealizadores, Tim Berners-Lee: "The original idea of the web was that it should be a collaborative space where you can communicate through sharing information."

3 Tradução livre nossa para: "geographies of participation."

4 Ver Collier e Bear (2012), Cohen (2011) e Cassel (2011), entre outras.

1.2 Informações sobre o Data Set

O conjunto de dados em questão é usado para estimar o número de mulheres editoras e suas práticas de edição na Wikipedia espanhola, os quais foram extraídos pelos autores de um carregamento, datado de outubro de 2017, contendo 90.827.797 páginas de editores que permitiram a identificação de 963.591 editores registrados. Destes, focou-se apenas nos editores ativos, ou seja: com ao menos 50 edições; que estiveram ativos durante os últimos 5 anos; e que tivessem feito ao menos 1 publicação desde 01 de janeiro de 2012. Na Wikipedia espanhola, isto totaliza 28.763 editores.

Como o interesse era analisar as informações que os editores divulgam sobre si mesmos por meio de sua página pessoal, foram filtrados os editores que não haviam desenvolvido sua própria página, deixando um total de 13.210 editores. Após um procedimento de seleção, foi gerado um "pacote" contendo até 55 perfis de usuário que foram atribuídos aos codificadores. Foram gerados 103 pacotes, totalizando 5.651 perfis de usuário a serem codificados.

Após um procedimento de codificação manual, foram codificados 4.746 de 5.651 páginas pessoais, após a exclusão de páginas de usuário vazias, removidas e bloqueadas.

1.3 Sobre os dados

Segundo as análises dos dados feitas por Minguillón et al. (2021):

- das 4.746 páginas pessoais, 2.029 eram homens (42,8%), 295 eram mulheres (6,2%) e 2.422 não puderam ser identificados e, portanto, permaneceram “desconhecidos” (51,0%);
- se forem levados em consideração apenas os perfis de usuário atribuíveis a mulheres e homens, aqueles identificados como mulheres representarão 12,9% do número total de editores.

Os editores podem especificar seu gênero nas configurações de preferência de sua conta na Wikipedia. A configuração de gênero é uma informação pública que pode ser extraída por meio da API Mediawiki. Quando comparados os resultados da extração manual de gênero para os 4.746 perfis codificados anteriormente com os gêneros da API Media wiki, descobriu-se que:

- Existem 2.792 usuários identificados como homens, compreendendo 58,8% do número total de editores.
- 353 usuários identificados como mulheres (172+58+123, em negrito e itálico), compreendendo 7,4%.
- Os outros 1.601 usuários permanecem rotulados como “desconhecidos” (33,8%).

Esta análise se propõe a investigar, utilizando a metodologia CRISP-DM: *qual o provável gênero, dentre os presentes no dataset, dos editores da wikipedia?*

2. Entendimento dos dados (análise exploratória)

Para fazer uma exploração inicial da base de dados, foram mantidos todos os atributos existentes:

Atributo	Descrição
gender	0 (unknown), 1 (male), 2 (female)
C_api	gêneros extraídos da API WikiMedia, codificados como female / male / unknown
C_man	gêneros extraídos da codificação de conteúdo, codificado como 1 (male) / 2 (female) / 3 (unknown)
E_NEds	I índice de estrato IJ (0,1,2,3)
E_Bpag	J índice de estrato IJ (0,1,2,3)
firstDay	primeira edição na Wikipedia espanhola (YYYYMMDDHHMMSS)
lastDay	última edição na Wikipedia espanhola (YYYYMMDDHHMMSS)
NEds	número total de edições
NDays	número de dias (lastDay-firstDay+1)
NActDays	número de dias com edições
NPages	número de páginas diferentes editadas
NPcreated:	número de páginas criadas
pagesWomen	número de edições em páginas relacionadas a mulheres
wikiprojWomen	número de edições no WikiProjects relacionados a mulheres
ns_user	número de edições no espaço de usuário
ns_wikipedia	número de edições no namespace wikipedia
ns_talk	número de edições no espaço de discussões
ns_userTalk	número de edições em espaço de discussões dos usuários
ns_content	número de edições em páginas de conteúdo
weightIJ	corrigindo peso do estrato IJ
NIJ	número de elementos em estrato IJ

Exceto por gender, C_api e C_man, os atributos do dataset são numéricos. Uma descrição inicial de algumas métricas, a partir do uso do método `.describe()`, demonstra que os atributos concernentes aos números de edições têm média bastante divergente dos valores máximos, para cada um deles. Assim, os valores inesperados representam comportamentos dissonantes de editores muito ativos na plataforma de Wiki.

	E_NEds	E_Bpag	firstDay	lastDay	NEds	NDays	NActDays	NPag
count	4746.000000	4746.000000	4.746000e+03	4.746000e+03	4746.000000	4746.000000	4746.000000	4746.0000
mean	1.484197	1.646228	2.009942e+13	2.015489e+13	2029.969448	2036.607880	183.162663	689.4519
std	1.099795	1.079263	3.516337e+10	1.748104e+10	7793.300833	1336.119914	374.034481	3355.3024
min	0.000000	0.000000	2.002011e+13	2.012010e+13	50.000000	1.000000	1.000000	1.0000
25%	1.000000	1.000000	2.007042e+13	2.014070e+13	95.000000	835.250000	24.000000	29.0000
50%	1.000000	2.000000	2.009121e+13	2.016072e+13	218.000000	2035.500000	53.000000	68.0000
75%	2.000000	3.000000	2.013040e+13	2.017073e+13	757.750000	3146.500000	154.000000	219.7500
max	3.000000	3.000000	2.017093e+13	2.017100e+13	153193.000000	5349.000000	3843.000000	94142.0000

Figura 1: Resultado da aplicação do método `.describe()`

Os dados estão rotulados de acordo com gênero dos editores e, por isso, avalia-se a distribuição de cada atributo de acordo com o rótulo atribuído aos editores. Para a criação dos gráficos, considerou-se apenas os valores do atributo ‘gender’, pois foram atribuídos pelos pesquisadores após avaliação qualitativa dos perfis estudados.

2.1 Quantidade de editores por gênero

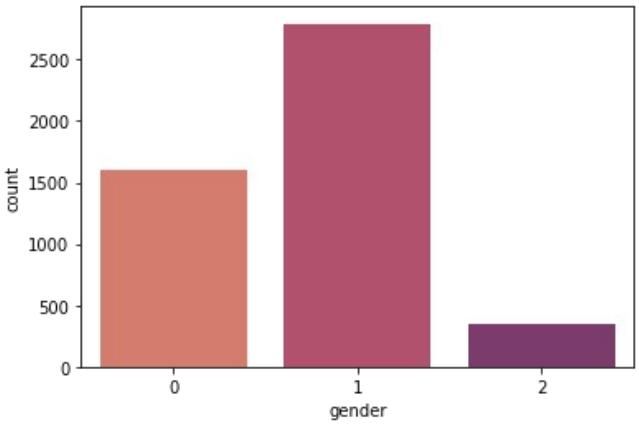

Figura 2: Quantidade de editores por gênero

Como é possível notar no gráfico da Figura 1, a quantidade de editores do gênero 1 - masculino é consideravelmente maior que a quantidade de editores do gênero 2 - feminino, e a quantidade de pessoas de gênero 0 - desconhecido se encontra em um valor intermediário, porém muito significante.

2.2 Matriz de correlação

Com a utilização do método `.corr()` para tratamento de dataframes da biblioteca pandas, criamos a matriz, cuja versão gráfica foi gerada com uso do método `.heatmap()` da biblioteca Seaborn.

À primeira vista, percebe-se que vários dos atributos possuem relacionamentos significativos entre si, graças à presença do vermelho em diversas correlações em vários tons. A visualização indica ser mais forte a relação entre o par de atributos NEds (número total de edições) e ns_content (número de edições no conteúdo páginas), dada a cor vermelha mais fechada. Neste caso, ela representa o valor aproximado de 0.98, muito próximo de 1, indicador de uma correlação substancial (ver Figuras 3 e 4).

As observações iniciais indicam também a pouca relevância dos atributos E_Neds, E_Bpag, firstDay, lastDay: a tonalidade mais clara indica valores próximos de zero ou abaixo de zero ao longo de toda a linha vertical, por exemplo. O mesmo ocorre para wheightIJ e NIJ, os quais apenas têm relação forte entre si. Estes atributos foram gerados como métricas a partir do dataset coletado, por isso faz sentido não dialogarem com o comportamento dos demais.

Uma outra relação significativa se estabelece entre NPages e NEds, de aproximadamente 0.90, o que faz sentido dado que o número total de edições depende do número de páginas diferentes editadas. Destacam-se, ainda, as relações entre atributos: com valores maiores que 0.8 (ns_content e NPages (~0.88)), maiores que 0.7 (NactDays e NEds (~0.78), ns_content e nActDays (~0.76), ns_wikipedia e ns_userTalk (0.74), ns_talk e ns_wikipedia (0.74), ns_userTalk e ns_talk (0.71)), maiores que 0.6 (nPages e NactiveDays (0.66), nactDays e ns_talk(0.60), ns_talk e nEds (0.61)) e maiores que 0.5 (n_userTalk e nactiveDays (0.51), nactDays e nEds (0.52), nEds e nPCreated (0.53), n_userTalk e nPages (0.56), ns_talk e nPages (0.57), np_created e nPages (0.59) e ns_content e np_created(0.55)) (ver Figura 4).

Os números fazem sentido pois são mais correlacionados atributos relativos a quantidade de páginas editadas, de páginas criadas, de interações nos espaços de discussão e de atividade na plataforma Wikipedia. O avanço da investigação permitirá descobrir se, além de quais atributos estão bastante correlacionados, quais são determinantes para a classificação de editores por gênero.

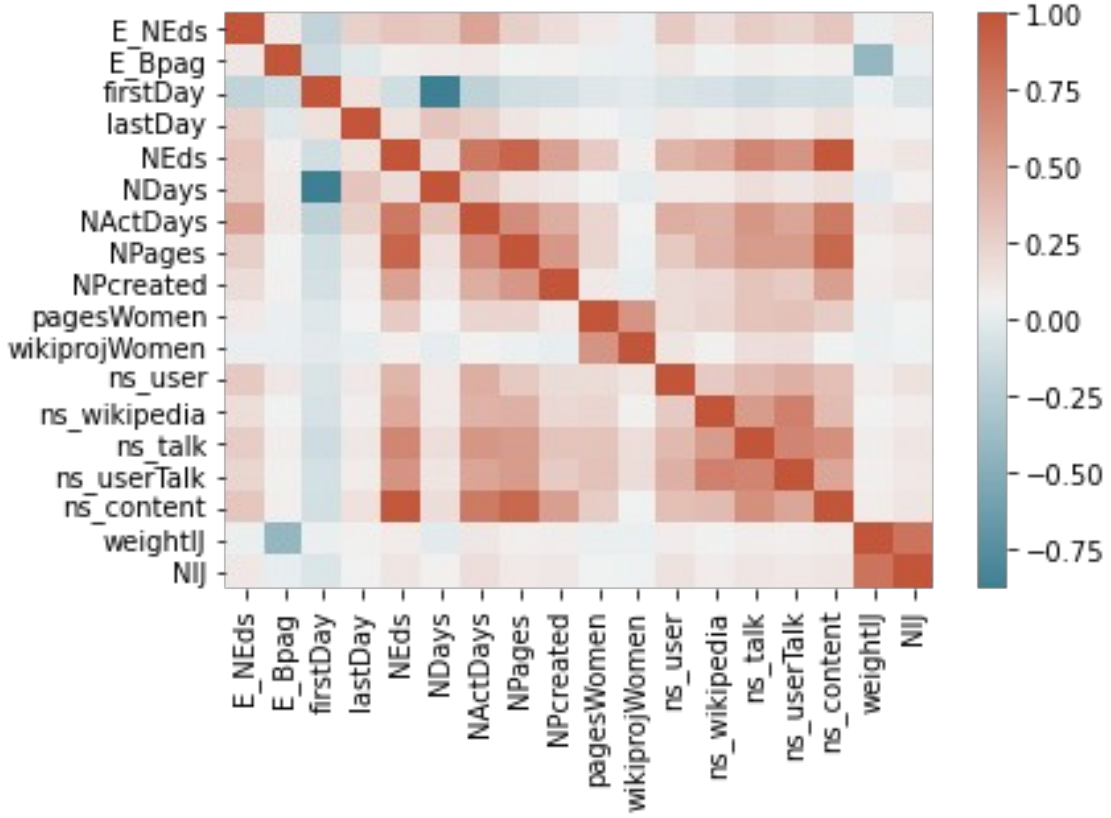

Figura 3: Matriz de Correlação com mapa de cores

	E_NEds	E_Bpag	firstDay	lastDay	NEds	NDays	NActDays	NPages	NPcreated	pagesWomen	wikiprojWomen	ns_user	ns_wikipedia	ns_talk	ns_userTalk	ns_content	weightIJ	NIJ
E_NEds	1.000000	0.127476	-0.186356	0.259000	0.330999	0.311284	0.524455	0.260731	0.181841	0.106484	0.030551	0.298505	0.174467	0.279854	0.225501	0.318804	0.040102	0.115894
E_Bpag	0.127476	1.000000	-0.126777	-0.025059	0.085689	0.108217	0.120005	0.059046	0.077025	0.038336	0.027241	0.127708	0.055520	0.086462	0.074981	0.079052	-0.417940	0.021281
firstDay	-0.186356	-0.126777	1.000000	0.161636	-0.099369	-0.875006	-0.199639	-0.100343	-0.085885	-0.030236	-0.005189	-0.048975	-0.068091	-0.113621	-0.085746	-0.095595	0.034635	-0.042541
lastDay	0.259000	-0.025059	0.161636	1.000000	0.163663	0.327911	0.253744	0.135295	0.084728	0.063984	0.022284	0.120089	0.085469	0.125636	0.104564	0.159510	0.069976	0.062573
NEds	0.330999	0.085689	-0.099369	0.163663	1.000000	0.180513	0.787403	0.901843	0.538389	0.295441	0.077223	0.426962	0.487828	0.693306	0.615990	0.983966	0.105281	0.142788
NDays	0.311284	0.108217	-0.875006	0.327911	0.180513	1.000000	0.323359	0.166512	0.127290	0.063258	0.017077	0.107488	0.110421	0.174524	0.137004	0.174822	0.001911	0.071814
NActDays	0.524455	0.120005	-0.199639	0.253744	0.787403	0.323359	1.000000	0.660793	0.465650	0.236907	0.064494	0.468809	0.436626	0.603206	0.519505	0.769303	0.125003	0.177994
NPages	0.260731	0.059046	-0.100343	0.135295	0.901843	0.166512	0.660793	1.000000	0.591040	0.235077	0.041384	0.308889	0.451903	0.573245	0.568630	0.883104	0.081657	0.107028
NPcreated	0.181841	0.077025	-0.085885	0.084728	0.538389	0.127290	0.465650	0.591040	1.000000	0.112592	0.021538	0.190143	0.209787	0.337493	0.290529	0.553770	0.086729	0.126870
pagesWomen	0.106484	0.038336	-0.030236	0.063984	0.295441	0.063258	0.236907	0.235077	0.112592	1.000000	0.616682	0.187476	0.233701	0.336585	0.349960	0.280602	0.036831	0.056643
wikiprojWomen	0.030551	0.027241	-0.005189	0.022284	0.077223	0.017077	0.064494	0.041384	0.021538	0.616682	1.000000	0.135937	0.072416	0.175062	0.184847	0.056959	0.033497	0.050328
ns_user	0.298505	0.127708	-0.048975	0.120089	0.426962	0.107488	0.468809	0.308889	0.190143	0.187476	0.135937	1.000000	0.300814	0.398770	0.443839	0.366414	0.094005	0.155931
ns_wikipedia	0.174467	0.055520	-0.068091	0.085469	0.487828	0.110421	0.436626	0.451903	0.209787	0.233701	0.072416	0.300814	1.000000	0.570182	0.742488	0.383592	0.067499	0.093849
ns_talk	0.279854	0.086462	-0.113621	0.125636	0.693306	0.174524	0.603206	0.573245	0.337493	0.336585	0.175062	0.398770	0.570182	1.000000	0.712549	0.643619	0.091385	0.136182
ns_userTalk	0.225501	0.074981	-0.085746	0.104564	0.615990	0.137004	0.519505	0.568630	0.290529	0.349960	0.184847	0.443839	0.742488	0.712549	1.000000	0.522963	0.082268	0.119187
ns_content	0.318804	0.079052	-0.095595	0.159510	0.983966	0.174822	0.769303	0.883104	0.553770	0.280602	0.056959	0.366414	0.383592	0.643619	0.522963	1.000000	0.100263	0.134885
weightIJ	0.040102	-0.417940	0.034635	0.069976	0.105281	0.001911	0.125003	0.081657	0.086729	0.036831	0.033497	0.094005	0.067499	0.091385	0.082268	0.100263	1.000000	0.812083
NIJ	0.115894	0.021281	-0.042541	0.062573	0.142788	0.071814	0.177994	0.107028	0.126870	0.056643	0.050328	0.155931	0.093849	0.136182	0.119187	0.134885	0.812083	1.000000

Figura 4: Matriz de Correlação com valores

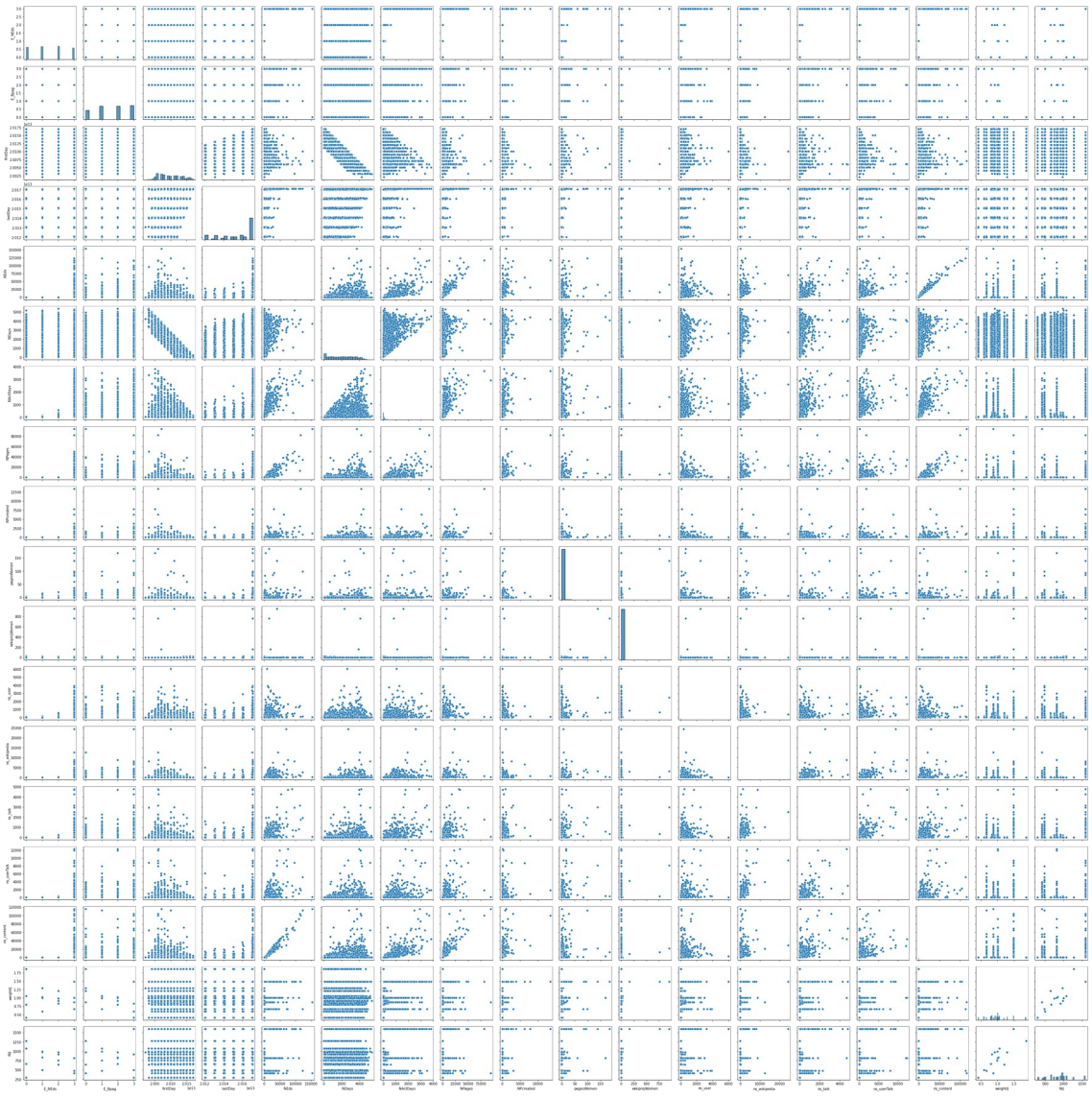

Figura 5: Visualização em pares das relações entre atibutos

O pairplot reforça o que foi observado na matriz de correlação. É possível observar que os pares de atributos que possuem maior correlação apresentam gráficos com maior dispersão.

2.3 Outras análises

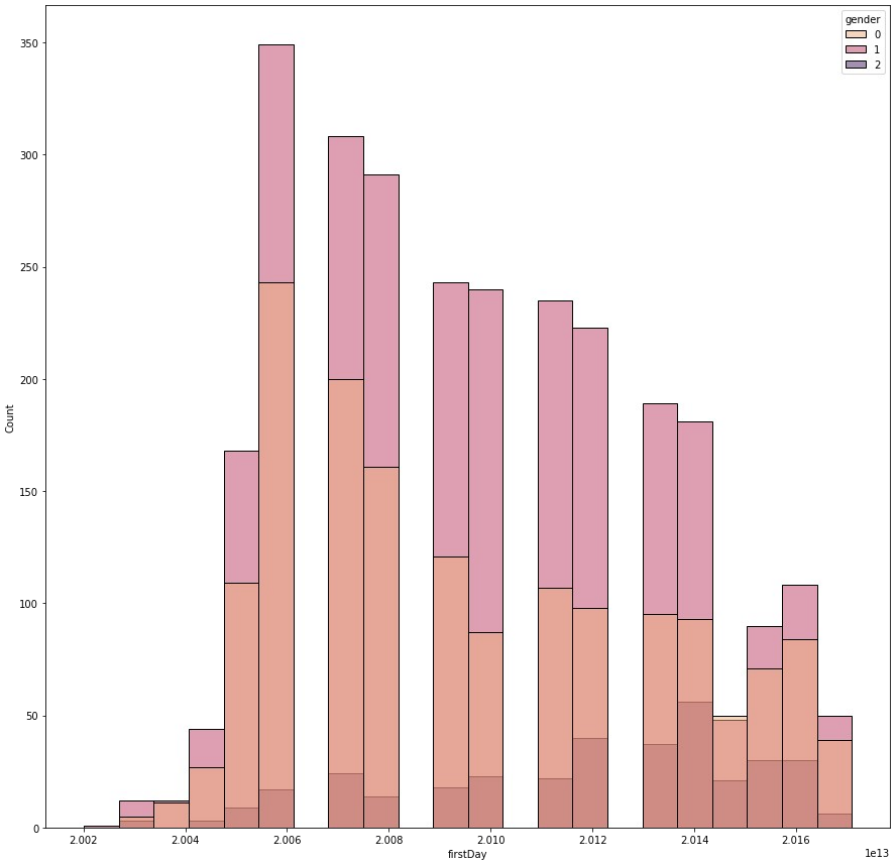

Figura 6: Primeiro dia de edições na Wikipédia

Como pode ser observado, houve uma maior quantidade de editores ingressantes no ano de 2006. A partir de então, os índices de ingresso caíram pouco a pouco e entre 2014 e 2015 ocorreu uma queda mais significativa, voltando a subir após 2015, mas caindo novamente após 2016.

As proporções entre quantidade de editores de gêneros diferentes se manteve praticamente a mesma na maioria dos anos, exceto nos períodos entre 2002 e 2004 e entre 2014 e 2015, onde a proporção de editores do gênero masculino não foi muito maior do que a soma dos demais gêneros.

Além disso, é possível observar um aumento no ingresso de mulheres no ano de 2014 com relação aos outros anos.

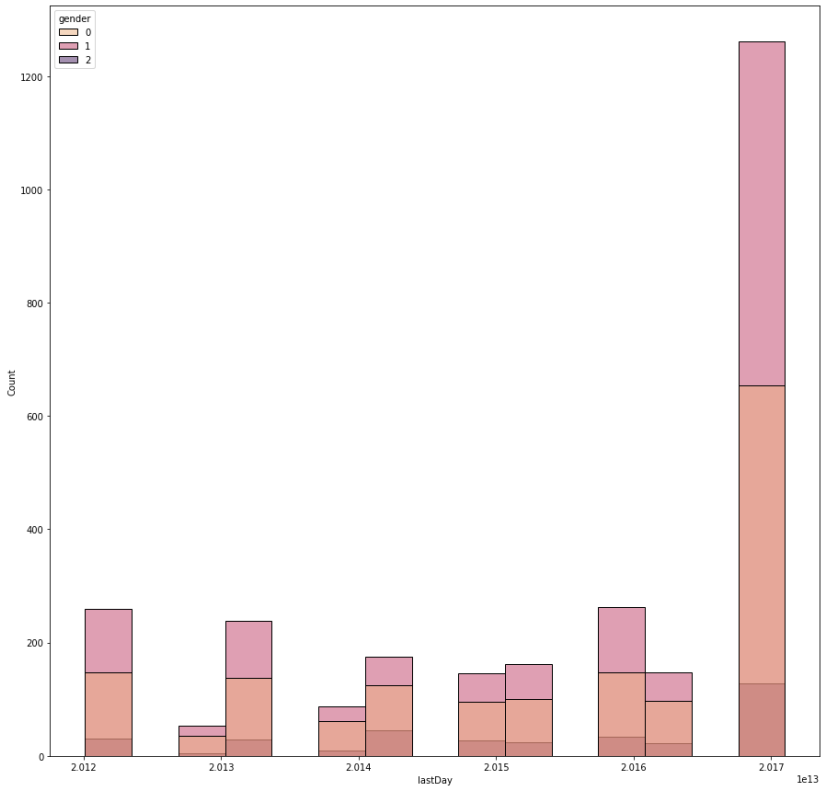

Figura 7: Último dia de edições feitas na Wikipédia

Na Figura 4, podemos verificar que a maior quantidade de últimas edições realizadas se concentra no ano de 2017, provavelmente devido à seleção dos dados/amostras ter sido realizada nesse ano.

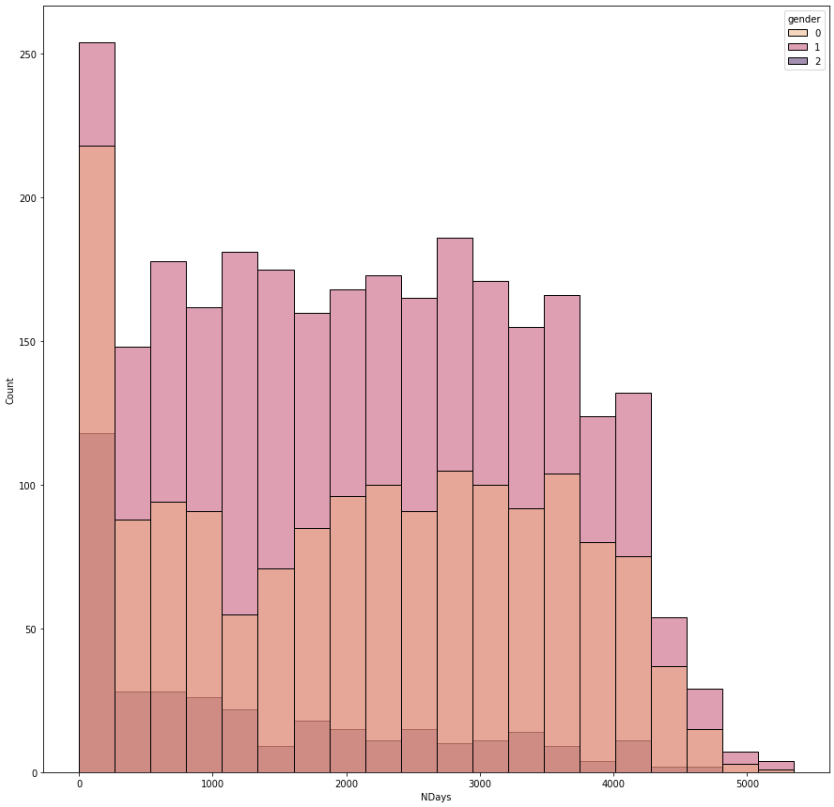

Figura 8: Número de dias desde a entrada na Wikipedia (último dia - primeiro dia + 1)

A quantidade de editores vai diminuindo à medida que a quantidade de dias aumenta. Isso quer dizer que a maioria dos editores esteve ativo por poucos dias e apenas uma parcela deles permaneceu ativa nos dias posteriores, sendo que apenas uma pequena quantidade (menos de 10) permaneceu ativa por mais de 5000 dias.

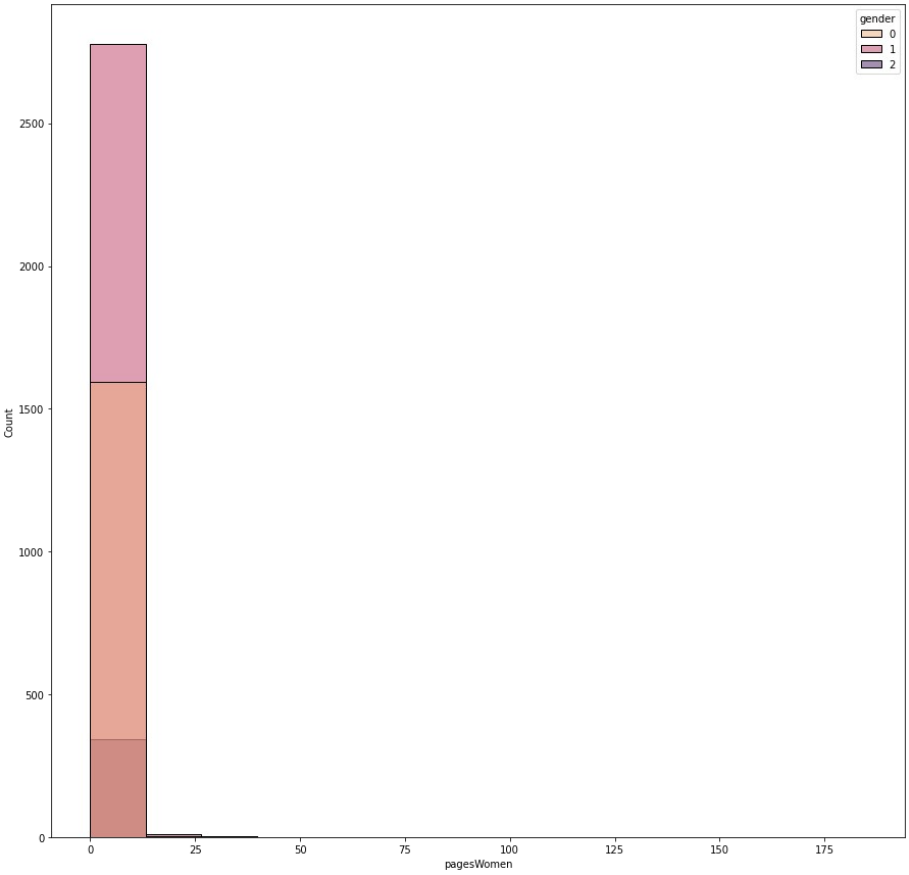

Figura 9: Número de edições em páginas relacionadas a mulheres

Na Figura 13 é possível notar uma prevalência de editores que se identificam com o gênero masculino ou que apresentam gênero desconhecido até mesmo em páginas relacionadas a mulheres.

É possível deduzir, analisando o gráfico, que mesmo que as pessoas com gêneros desconhecidos se declarassem do gênero feminino, a quantidade de editoras mulheres permaneceria inferior à de editores homens.

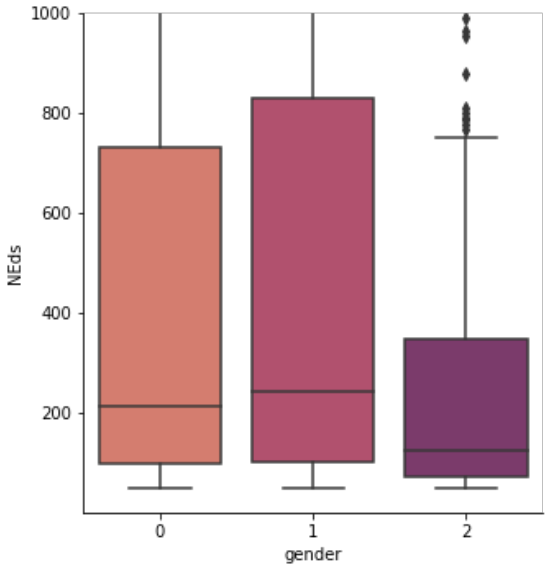

Figura 10: Número de edilões por gênero

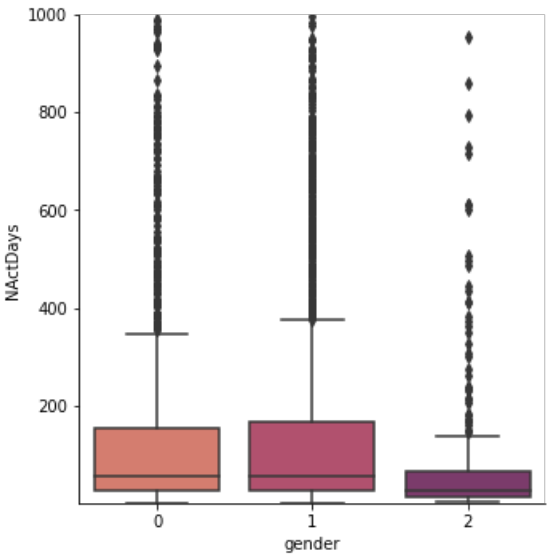

Figura 11: Número de dias ativos por gênero

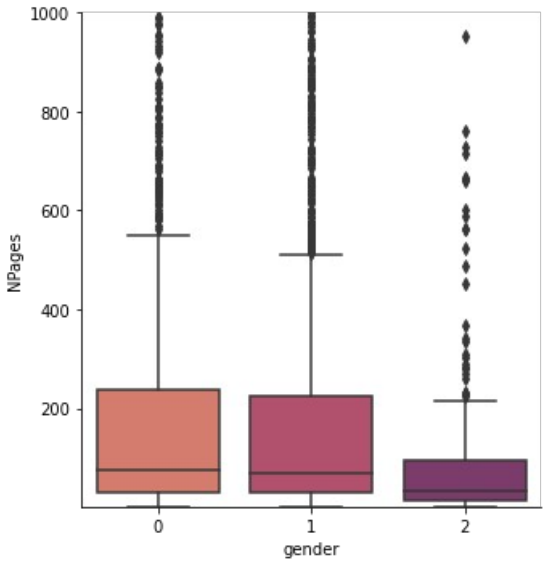

Figura 12: Número de páginas editadas

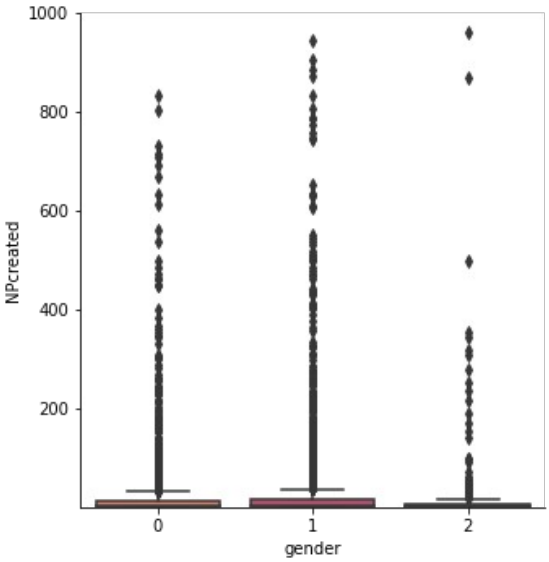

Figura 13: Número de páginas criadas

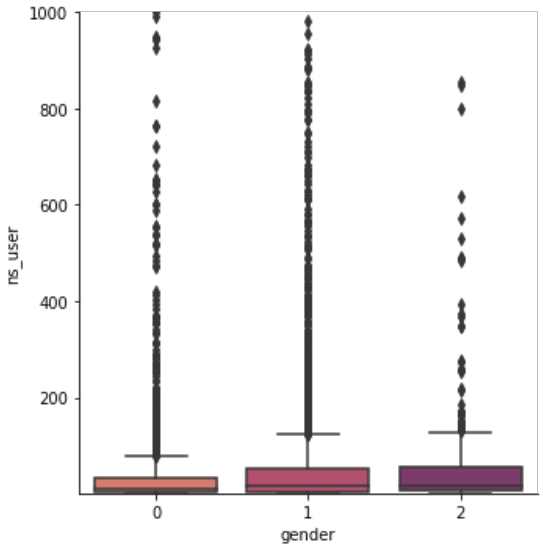

Figura 14: Número de edições no namespace do usuário

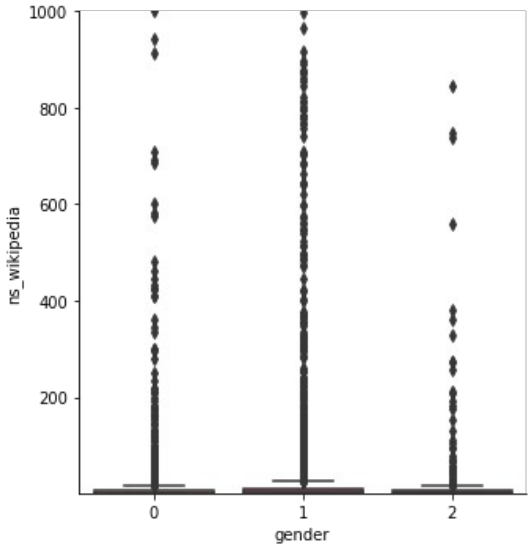

Figura 15: Número de edições no namespace da Wikipedia

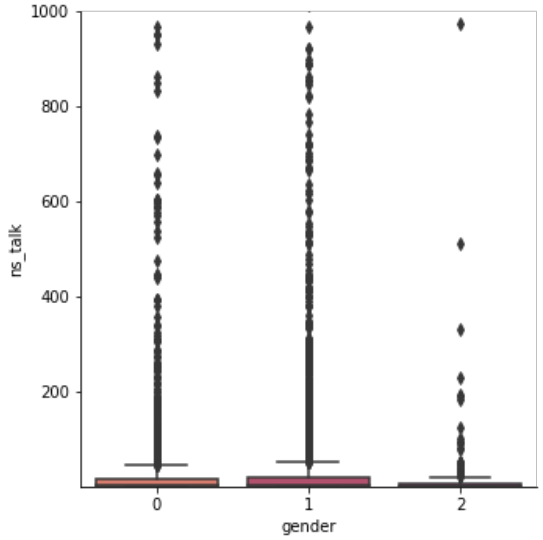

Figura 16: Número de edições no namespace talk

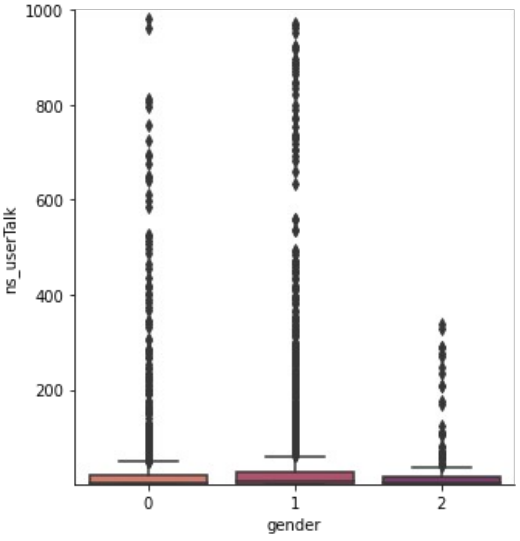

Figura 17: Número de edições no namespace usaer talk

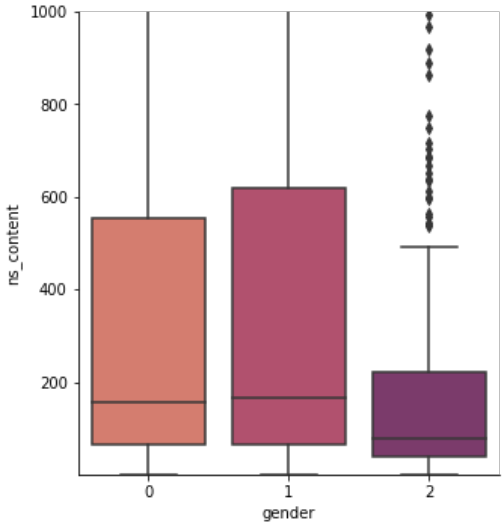

Figura 18: Número de edições nas páginas de conteúdo

Nos gráficos das figuras, reproduz-se a disparidade de gênero, pois o número de dias ativos é substancialmente maior para usuários homens e de pessoas de gênero desconhecido. Nos gráficos das figuras 5, 6 e 7, nota-se que 75% de editores homens e de gênero desconhecido realizam, mais que o dobro de edições, têm mais que o dobro de dias ativos e de páginas editadas que as mulheres.

O número de páginas criadas e os números de edição dos namespaces (figuras de 8 a 12) são baixos em comparação aos de edição de páginas. Mas as relações são similares, exceto para o número de edições no namespace do usuário, mais equânime para homens e mulheres. Neste caso, é possível observar uma maior quantidade de outliers entre os editores do gênero masculino. No gráfico da figura 13, contudo, percebe-se a repetição do já descrito padrão de contribuições menores por parte das editoras nas páginas de conteúdo.

Alguns atributos não tiveram os gráficos analisados, por terem sido considerados dispensáveis para a pesquisa: referem-se aos estratos que originaram a amostra da qual dispomos. São eles:

- **E_Neds** (índice I do estrato IJ) - referência ao índice I dos estratos que originaram a amostra de dados da qual dispomos;
- **E_Bpag** (índice J do estrato IJ) - eferência ao índice J dos estratos que originaram a amostra de dados da qual dispomos;
- **weightIJ** (Correção do peso para o estrato IJ) - pesos amostrais para cada amostra/estrato, determinados pelo número de edições e pelo tamanho da página do usuário para os 4.746 perfis codificados.

- **NIJ** (número de elementos do estrato) - quantidade de elementos da amostra do qual determinado valor foi retirado.

Pelo exposto, a análise exploratória possibilitou identificar que alguns dos atributos do dataset são pouco relevantes para o intuito da presente pesquisa, principalmente os atributos que tratam-se de referências aos estratos que originaram a amostra de dados da qual dispomos, podendo ser necessária sua remoção.

3. Preparação dos dados

Para esta etapa, o primeiro passo foi a definição de atributos, feita com mescla de entendimento do problema para determinar seleções manuais e a seleção automática com aplicação do modelo de Regressão Linear. Mais especificamente, aplicou-se o 'lreg' da biblioteca SciKitLearn. Decidimos pela métrica 'Mean squared error' como hiperparâmetro de 'scoring' para calcular o erro quadrático médio de predição.

3.1 Sobre a seleção de atributos

Esta etapa teve de ser refeita algumas vezes pois os resultados da primeira combinação de escolha manual e automática, que obteve o conjunto "firstDay", "lastDay", "NEds", "NDays", "NActDays", "NPages", "NPcreated", "pagesWomen", "ns_wikipedia", "ns_talk", "ns_userTalk", "ns_content", resultou em um comportamento incomum ao gerarmos as primeiras Árvores de Decisão. No primeiro experimento, haviam sido retirados manualmente os atributos de matriz e pesos criados pelos coletores do dataset; com subsequente seleção automática via Regressão Linear dentre os atributos restantes. A árvore resultante apenas considerava como parâmetros relevantes à classificação 'frstDay' e 'lastDay'. Este comportamento não é coerente com a investigação baseada neste dataset, que contém análise multidisciplinar sobre as interações dos editores da Wikipedia. Os autores, na sessão 3, de análise dos resultados, explanam suas descobertas e em momento algum utilizam primeiro e último dia de edições como atributo relevante para a pesquisa (MINGUILLÓN et al. 2022).

Além da árvore inconsistente, ao se levar em consideração o conhecimento obtido nas fases de compreensão do problema (tópico 1) e dos dados (tópico 2), havia mais indícios de que estes atributos poderiam ser descartados, pois se relacionavam pouco com os demais – vide análise da matriz de correlação (tópico 2.2). É o caso dos

atributos relativos à matriz de pesos criada por Minguillón et al. (2022) e, novamente, os atributos firstDay e lastDay.

Para confirmar as suspeitas, optou-se por realizar testes de impacto das possíveis escolhas na performance dos modelos, para verificar se haveriam vantagens em incluir os atributos avaliados como dispensáveis, pois o algoritmo de seleção estava incluindo alguns deles em detrimento de outros que se entendia serem mais informativos.

```
clf_knn = neighbors.KNeighborsClassifier(n_neighbors=78, p=1)
clf_knn = clf_knn.fit(normalizedX_train, y_train)
print("\nAcuracia Treinamento",  clf_knn.score(normalizedX_train, y_train))

clf_knn = clf_knn.fit(normalizedX_trainp, y_train_p)
print("\nAcuracia Treinamento parcial" ,clf_knn.score(normalizedX_trainp, y_train_p))

clf_knn = clf_knn.fit(normalizedX_valid, y_valid)
print("\nAcuracia validação" ,clf_knn.score(normalizedX_valid, y_valid))
```

```
Acuracia Treinamento 0.5919388830347735

Acuracia Treinamento parcial 0.5974710221285564

Acuracia validação 0.5742887249736565
```

Figura 19: Print das acurácias da classificação com KNN

```
clf_knn = neighbors.KNeighborsClassifier(n_neighbors=78, p=1)
clf_knn = clf_knn.fit(normalizedBase, y)
print("\nAcuracia do Dataset Base",  clf_knn.score(normalizedBase, y))

clf_knn = clf_knn.fit(normalizedX_semdias, y)
print("\nAcuracia Sem LastDay e FirstDay" ,clf_knn.score(normalizedX_semdias, y))
```

```
Acuracia do Dataset Base 0.5939738727349346

Acuracia Sem LastDay e FirstDay 0.5910240202275601
```

Figura 20: Print das acurácias da classificação com KNN

```
clf_tree = DecisionTreeClassifier (
    criterion= 'gini',
    max_depth= 3,
    min_samples_leaf= 3,
    min_samples_split= 4,
    random_state = 10
    )
clf_tree = clf_tree.fit(normalizedX_train, y_train)
print("\nAcuracia Treinamento",  clf_tree.score(normalizedX_train, y_train))

clf_tree = clf_tree.fit(normalizedX_trainp, y_train_p)
print("\nAcuracia Treinamento parcial" , clf_tree.score(normalizedX_trainp, y_train_p))

clf_tree = clf_tree.fit(normalizedX_valid, y_valid)
print("\nAcuracia validação" , clf_tree.score(normalizedX_valid, y_valid))
```

```
Acuracia Treinamento 0.5903582718651211

Acuracia Treinamento parcial 0.5999297506146821

Acuracia validação 0.5795574288724974
```

Figura 21: Print das acurácias da classificação com árvore de decisão

```
[ ]  clf_tree = clf_tree.fit(normalizedBase, y)
     print("\nAcuracia Dataset Base" , clf_tree.score(normalizedBase, y))

     clf_tree = clf_tree.fit(normalizedX_semdias, y)
     print("\nAcuracia Dataset Sem FirstDay e LastDay" , clf_tree.score(normalizedX_semdias, y))

     clf_tree = clf_tree.fit(normalizedX_train, y_train)
     print("\nAcuracia Dataset de Treinamento selecionado (sem os IJ)" , clf_tree.score(normalizedX_train, y_train))
```

```
Acuracia Dataset Base 0.5882848714707122

Acuracia Dataset Sem FirstDay e LastDay 0.6011378002528445

Acuracia Dataset de Treinamento selecionado (sem os IJ) 0.5903582718651211
```

Figura 22: Print das acurácias da classificação com árvore de decisão

Como é possível notar, a diferença nas acurácias em ambos os classificadores é bastante sutil. No entanto, observamos um aumento da acurácia da classificação com árvore de decisão do dataset do qual foram removidos os atributos com IJ (que são referências às amostras de onde os dados do conjunto de dados foram selecionados) e os atributos Firstday e Lastday, que se referem aos dias em que os editores iniciaram e encerraram, respectivamente, suas atividades na plataforma wikipédia.

Decidiu-se remover manualmente os atributos de índices (IJ) e de dias (fistDay e lastDay): a acurácia similar nos leva a decidir com base no entendimento do domínio, e a pesquisa dos especialistas aliada aos resultados da matriz de correlação podem embasar esta decisão.

Escolheu-se o valor 10 para o parâmetro “k_features”. O número surgiu do

entendimento do problema e dos dados: todos os atributos concernentes à interação dos editores na plataforma dizem algo sobre o comportamento dos usuários. Desse modo, a análise o modelo precisaria considerar, minimamente, uma dezena de atributos – vide, novamente, matriz de correlação no tópico 2.2. O “forward = True” significa que a seleção automática está sendo feita pela seleção de recursos para frente (forward) e não através do método de seleção de recursos para trás (backward). Ao inicializar “verbose = 1”, permite-se imprimir o resumo do modelo a cada iteração. Decidimos pela métrica ‘Mean squared error’ como hiperparâmetro de ‘scoring’ para calcular o erro quadrático médio de predição. Aplicamos a Regressão linear com esses hiperparâmetros aos dados, mais especificamente o lreg da biblioteca SciKitLearn.

Deste processo, foram mantidos 'NEds', 'NDays', 'NActDays', 'NPages', 'NPcreated', 'wikiprojWomen', 'ns_wikipedia', 'ns_talk', 'ns_userTalk', 'ns_content'. Colocou-se, em seguida, os atributos selecionados em um novo Data Frame e redefinimos o conjunto “X” de dados com base nestes. Após a escolha dos atributos, dividimos a base de dados em base de treinamento e base de teste. Ainda há uma etapa de preparação a ser feita, a normalização. No entanto, é preferível executá-la após a separação para que se evite vazamento de dados da base de teste para a de treino. Ou seja, para que não haja, de fato, interferência dos valores presentes em uma base nas transformações executadas na outra.

Dentre os métodos considerados para realizar a normalização dos valores numéricos (normalização, aqui, usada no sentido genérico, sem diferenciação entre re-escala/standardização), escolhemos o .Normalizer(). É necessário normalizar pois, para o caso de métodos baseados em distância, evita-se conferir um peso maior a atributos com um intervalo grande que iriam suplantar atributos com intervalos pequenos (os quais, consequentemente, teriam um peso menor) (HAN, KAMBER e PEI; 2011). No caso deste dataset, percebe-se a existência de alguns atributos com intervalos grandes, passíveis de gerar esse viés no modelo a ser aplicado. Aplicamos a normalização para ambos os subconjuntos de treino e teste.

4. Modelagem

Para realizar a busca pelo melhor classificador, aplicamos os modelos estudados durante a disciplina ao conjunto preparado de dados. As implementações KNeighborsClassifier, MLPClassifier, DecisionTreeClassifier e SVC, da biblioteca SciKit Learn e LVQ, da biblioteca Neupy, receberam os conjuntos de treinamento completo, treinamento parcial e de validação.

Para finalizar este passo, utilizou-se o CrossValidation para identificar qual conjunto de valores dos hiperparâmetros de cada modelo apresenta maior acurácia (score=’accuracy’) no reconhecimento das classes definidas para o dataset.

Paralelamente, ajustou-se dicionários com valores de hiperparâmetros a serem aplicados ao otimizador HalvingRandomSearchCV. Após a busca randômica, foram retornados valores ótimos melhor ajuste de cada modelo.

4.1 K- Nearest Neighbors - KNN

Os valores de acurácia após aplicação da validação cruzada para este modelo indicam um comportamento desejável a partir de valores de k maiores que 20 ou 30 vizinhos, quando passamos a ter a acurácia próxima de 0.6.

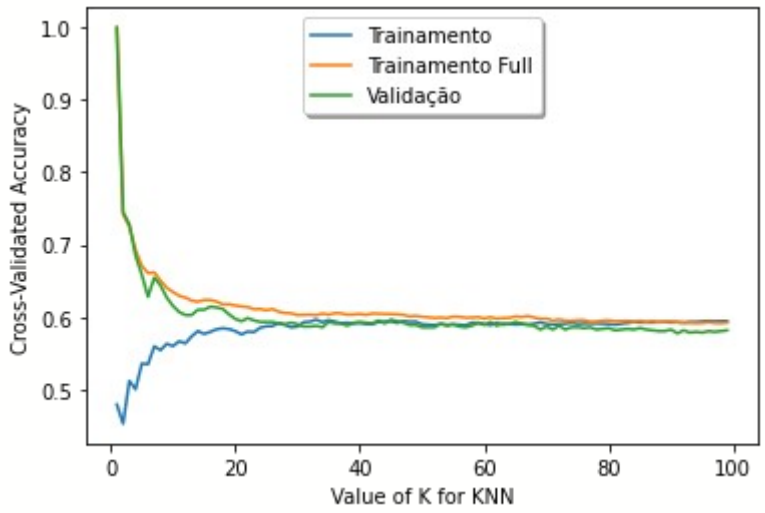

Figura 23: Valores de acurácia após aplicação da validação cruzada

A aplicação do HalvingRandomSearchCV retornou os valores de 98 para o número de vizinhos e 1 (Distância de Manhatan) para a medida de distância. Com os hiperparâmetros ajustados dessa maneira, obtêm-se uma acurácia de 0.592 para dados de treinamento e 0.593 para os dados base (sem alterações). Em vista do gráfico para a busca randômica, entende-se que a ‘Halving’ traz um resultado ótimo para o modelo.

4.2 Learning Vector Quantization - LVQ

A visualização do gráfico para o LVQ pode dar a impressão de que existe uma distância maior entre as curvas para o treinamento completo e treinamento parcial e a curva de validação. A observação mais atenta evidencia que os valores são, em verdade, próximos. Os platôs formados nas curvas de treinamento indicam valor de acurácia próximo a 0.59 e, no caso da curva de validação, as constantes ocorrem em diversos intervalos correspondentes aos de mesmo comportamento das outras curvas, com acurácia em torno de 0.57. Assim, para os valores entre 2 e 9 subclasses e de 18 em diante, o modelo apresentaria os melhores desempenhos.

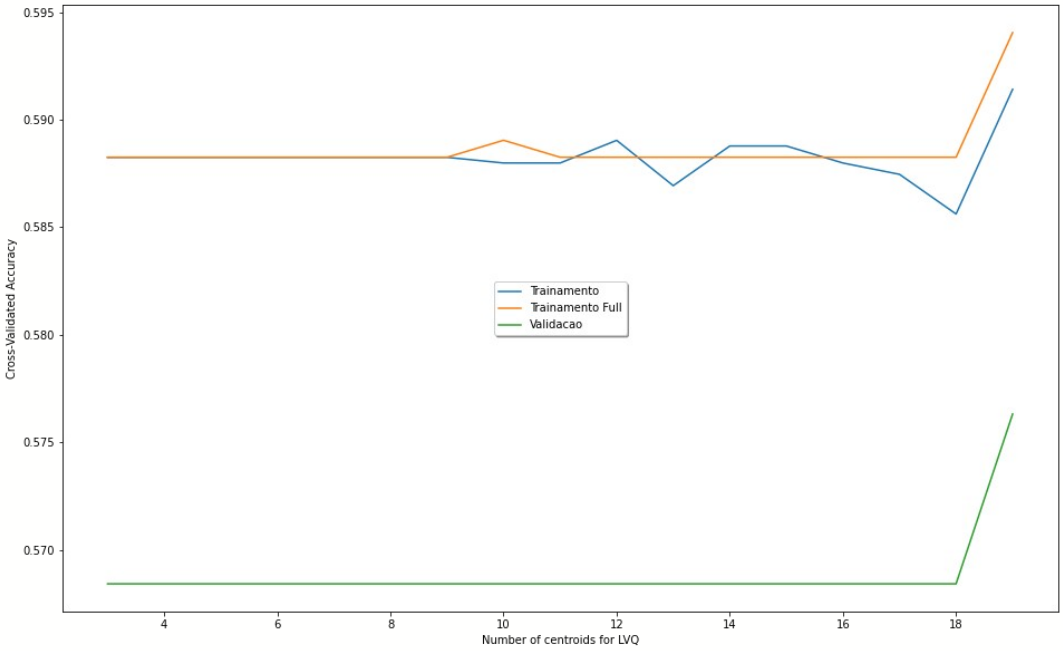

Figura 24*:* Valores de acurácia após aplicação da validação cruzada

Da aplicação do ‘HalvingRandomSearchCV’, obtém-se 23 para melhor valor de 'n_inputs', e 3 para 'n_subclasses'. Aplicando estes valores aos hiperparâmetros do

classificador, obtivemos as seguintes acurácias para os conjuntos de dados base, de treinamento parcial, de treinamento completo e de validação.

```
clf_lvq3 = algorithms.LVQ(n_inputs=23, n_subclasses=3, n_classes=3, weight=None)
clf_lvq = clf_lvq3.fit(normalizedX_trainp, y_train_p)
print("Acuracia de trainamento parcial: %0.4f" %  accuracy_score(clf_lvq.predict(normalizedX_trainp),y_train_p))

clf_lvq4 = algorithms.LVQ(n_inputs=23, n_subclasses=3, n_classes=3, weight=None)
clf_lvq = clf_lvq4.fit(normalizedX_valid, y_valid)
print("Acuracia de validação: %0.4f" %  accuracy_score(clf_lvq.predict(normalizedX_valid),y_valid))
```

```
Acuracia Baseline: 0.5883
Acuracia de trainamento: 0.5883
Acuracia de trainamento parcial: 0.5932
Acuracia de validação: 0.5697
```

Figura 25: Acurácias para aplicação do Halving search

Pelos valores gerados, os resultados da validação cruzada e do otimizador são consistentes, e é seguro sugerir a utilização do valor 3 para a quantidade de subclasses.

4.3 Support Vector Machine - SVM

Os dicionários de parâmetros contém valores para o parâmetro de regularização: C; o coeficiente do kernel: gamma; e o tipo de kernel a ser usado: rbf, polinomial, sigmóide ou linear.

Os melhores parâmetros para o classificador SVM foram:

```
[ ]  param_dist_svm = {
         'C': [0.1, 1, 10,],
         'gamma': [1, 0.1, 0.01, 0.001],
         'kernel': ['rbf','poly','sigmoid','linear']
     }

     rsh = HalvingRandomSearchCV(
         estimator=clf_svm, param_distributions=param_dist_svm, factor=2, random_state=10
     )
     rsh.fit(normalizedX_train, y_train)
     rsh.best_params_
```

```
{'C': 0.1, 'gamma': 0.01, 'kernel': 'sigmoid'}
```

Figura 26: Melhores parâmetros para o classificador SVM

O melhor desempenho do SVM foi com o kernel Sigmóide e valores de C=0.1 e gamma=0.01. O otimizador retorna valores coerentes com esta avaliação. A acurácia de

treinamento completo para esse caso é de 0.588, para treinamento parcial é de 0.593 e 0.568 para validação, como pode ser observado a seguir:

```
[49]  1 clf_svm = SVC(C= 0.1, gamma= 0.01, kernel= 'sigmoid')
      2
      3 clf_svm = clf_svm.fit(normalizedBase, ybase)
      4 print("\nAcuracia Do Dataset Base" ,clf_svm.score (normalizedBase, ybase))
      5
      6 clf_svm = clf_svm.fit(normalizedX_train, y_train)
      7 print("\nAcuracia De treinamento" ,clf_svm.score (normalizedX_train, y_train))
      8
      9 clf_svm = clf_svm.fit(normalizedX_trainp, y_train_p)
     10 print("\nAcuracia De treinamento parcial" ,clf_svm.score (normalizedX_trainp, y_train_p))
     11
     12 clf_svm = clf_svm.fit(normalizedX_valid, y_valid)
     13 print("\nAcuracia de validação" ,clf_svm.score (normalizedX_valid, y_valid))

Acuracia Do Dataset Base 0.5882848714707122

Acuracia De treinamento 0.5882507903055848

Acuracia De treinamento parcial 0.5932147562582345

Acuracia de validação 0.5684210526315789
```

Figura 27: Acurácias da classificação com SVM

É comum entre os resultados a escolha do kernel com função sigmóide e o valor de 0.001 para gamma.

4.4 Árvore de decisão

Para realizar a seleção de hiperparâmetros do classificador árvore de decisão foi utilizada a Successive Halving do Scikit Learn (HalvingRandomSearchCV), como na maioria dos demais classificadores. Por meio da '*halving*' obtivemos como melhores hiperparâmetros, isto é, a combinação que geraria um desempenho mais satisfatório do classificador: 'gini', como critério; max_depth com valor 6; min_samples_leaf = 2, e min_samples_split = 5.

Tal combinação, quando aplicada à função DecisionTreeClassifier (classificador árvore de decisão) gerou os seguintes valores de acurácia para os conjuntos de dados:

```
[52]  1 clf_tree = DecisionTreeClassifier (
      2     criterion= 'gini',
      3     max_depth= 6,
      4     min_samples_leaf= 2,
      5     min_samples_split= 5,
      6     random_state = 10
      7     )
      8
      9 clf_tree = clf_tree.fit(normalizedBase, ybase)
     10 print("\nAcuracia Dataset Base" , clf_tree.score(normalizedBase, ybase))
     11
     12 clf_tree = clf_tree.fit(normalizedX_train, y_train)
     13 print("\nAcuracia Treinamento",  clf_tree.score(normalizedX_train, y_train))
     14
     15 clf_tree = clf_tree.fit(normalizedX_trainp, y_train_p)
     16 print("\nAcuracia Treinamento parcial" , clf_tree.score(normalizedX_trainp, y_train_p))
     17
     18 clf_tree = clf_tree.fit(normalizedX_valid, y_valid)
     19 print("\nAcuracia validação" , clf_tree.score(normalizedX_valid, y_valid))

Acuracia Dataset Base 0.5941845764854614

Acuracia Treinamento 0.6172286617492097

Acuracia Treinamento parcial 0.6284584980237155

Acuracia validação 0.6802631578947368
```

Figura 28: Acurácias com árvore de decisão

Percebe-se valores elevados de acurácia do modelo para as bases, especialmente para a base de validação (0.68), superior às dos modelos já analisados. Foram geradas, então, as árvores de decisão para o conjunto de dados base e para o conjunto de treinamento, conforme pode ser observado nas imagens em dos anexos 1 e 2 e no notebook entregue em conjunto com este relatório.

A árvore com os dados de base reproduz o comportamento descrito na seção 3.1, Escolha dos atributos. Nessa árvore, apenas os atributos lastDay e firstDay são levados em conta para realizar a classificação, desde o tronco até as folhas. Já a árvore gerada a partir do dataframe reduzido após a seleção de atributos, apresenta comportamento mais coerente, e leva em consideração primeiramente o atributo WikiProjWomen, de medida da contribuição em projetos relativos à mulheres e, em seguida, atributos relativos às atividades de edição na plataforma, tais como nEds, nDays, nPages, nActiveDays, ns_userTalk, ns_content, ns_wikipedia, ns_content, Npcreated.

4.5 Neural Networks (MLP Classifier)

Há um curioso paralelismo visual entre as curvas que relacionam a acurácia dos treinamentos para as bases completa, parcial e de validação e o número de nós em uma camada. Do uso do RandomizedSearchCV, obtém-se o melhor score de aproximadamente 0.59 para os hiperparâmetros activation = ‘logistic’ e hidden_layer_sizes = ‘40’.

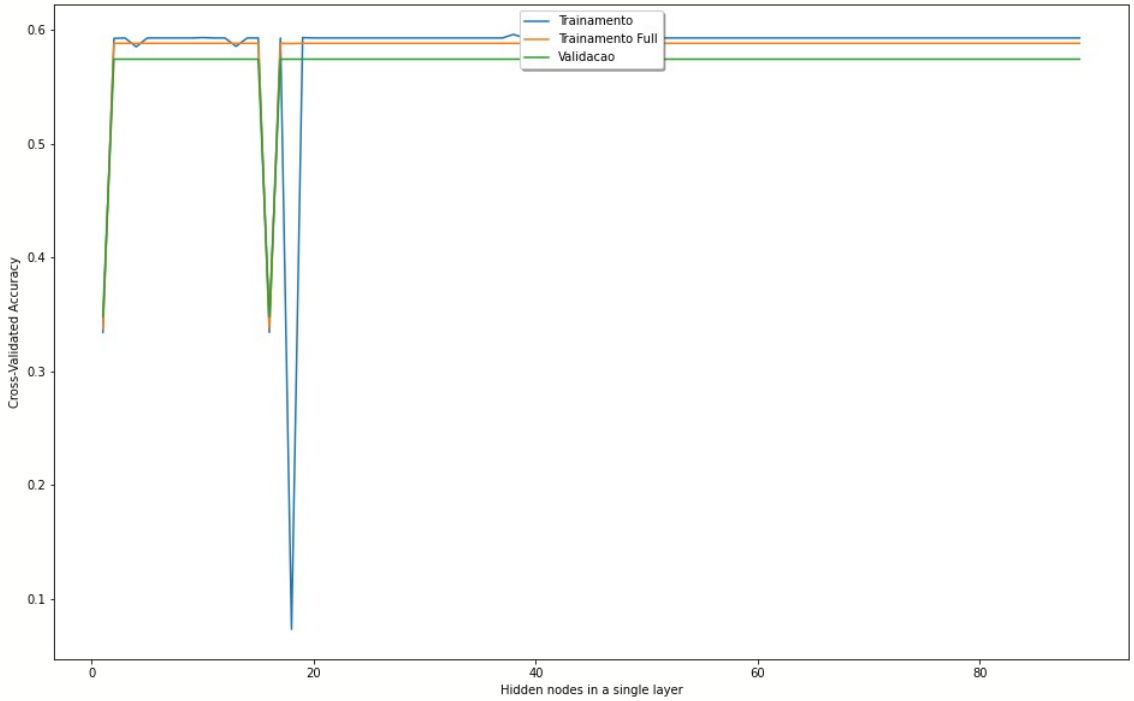

Figura 29: Acurácias das redes neurais

Quando aplica-se o otimizador ‘Halving’, tem-se, para a função de ativação regressão logística com 20 camadas ocultas, uma acurácia que pode ser observada na imagem abaixo para os dados base, de treinamento completos, parciais e de validação.

```
[58]  1 #clf_rn = MLPClassifier(activation= 'tanh', hidden_layer_sizes= 30)
      2 clf_rn = MLPClassifier(activation= 'logistic', hidden_layer_sizes= 60)
      3
      4 clf_rn = clf_rn.fit(normalizedBase, ybase)
      5 print("\nAcuracia Do Dataset Base" ,clf_rn.score (normalizedBase, ybase))
      6
      7 clf_rn = clf_rn.fit(normalizedX_train, y_train)
      8 print("\nAcuracia De treinamento" ,clf_rn.score (normalizedX_train, y_train))
      9
     10 clf_rn = clf_rn.fit(normalizedX_trainp, y_train_p)
     11 print("\nAcuracia De treinamento parcial" ,clf_rn.score (normalizedX_trainp, y_train_p))
     12
     13 clf_rn = clf_rn.fit(normalizedX_valid, y_valid)
     14 print("\nAcuracia de validação" ,clf_rn.score (normalizedX_valid, y_valid))

Acuracia Do Dataset Base 0.5882848714707122

Acuracia De treinamento 0.5882507903055848

Acuracia De treinamento parcial 0.5932147562582345

Acuracia de validação 0.5684210526315789
```

Figura 30: Acurácias para o MLPClassifier

4.6 Comitês de classificadores

Para todos os comitês, foram utilizados os hiperparâmetros encontrados atravpes da busca resultante da aplicação do HalvingRandomSearchCV.

4.6.1 AdaBoost

Os modelos usados no Adaboost foram KNN, Árvore de Decisão, Rede Neural e SVM, inicializados com os melhores hiperparâmetros selecionados pela HalvingRandomSearchCV anteriormente para cada um deles.

Como é possível observar na figura 31, o ensemble não imprime o seu desempenho em comparação ao desempenho do KNN e Rede neural, respectivamente. Este comportamento se deve a este comitê não dar suporte ao uso destes modelos.

Analisando a comparação entre o desempenho do modelo árvore de decisão individualmente com o desempenho do Adaboost, podemos notar que o menor desemprenho do classificador (0.59) foi superior ao do comitê (0.56).

A ordem de apresentação dos resultados é a seguinte: Árvore de decisão, KNN, Redes neurais e SVM.

```
Media clf  0.5836633663366336 Desvio  0.018397347659224607 Media Boosting 0.5635834636095188 Desvio  0.018698959403077923
Media clf  0.5922290255341324 Desvio  0.0054526744299609325 Media Boosting nan Desvio  nan
Media clf  0.5932158676393955 Desvio  0.0011730690532326439 Media Boosting nan Desvio  nan
Media clf  0.5932158676393955 Desvio  0.0011730690532326439 Media Boosting 0.5932158676393955 Desvio  0.0011730690532326439
```

Figura 31: Médias do AdaBoost

4.6.2 Bagging

No Bagging conseguimos incluir o LVQ e obtivemos os resultados dos desempenhos de quase todos os classifiers, exceto para o LVQ como classificador-base, para cuja implementação escolhida a aplicação desta métrica não é possível.

```
Media clf  0.5922290255341324 Desvio  0.0054526744299609325 Media Bagging 0.5932180389091541 Desvio  0.003233833700763366
Media clf  0.5932158676393955 Desvio  0.0011730690532326439 Media Bagging 0.5932158676393955 Desvio  0.0011730690532326439
Media clf  0.5932158676393955 Desvio  0.0011730690532326439 Media Bagging 0.5932158676393955 Desvio  0.0011730690532326439
Media clf  0.5836633663366336 Desvio  0.018397347659224607 Media Bagging 0.5912356696195935 Desvio  0.010356835495608832
Media clf  nan Desvio  nan Media Bagging 0.5932158676393955 Desvio  0.0011730690532326439
```

Figura 32: Médias do Bagging

A saída da execução nos mostra que o Bagging teve desempenho ainda inferior ao da árvore de decisão para os dados de treino (0.61) e ligeriamente melhor que o do SVM (0.583), e apresentou desempenhos exatamente similares aos do KNN e da rede neural, tanto a média dos desempenhos, quanto o desvio padrão.

4.6. 3 Voting

Para o Voting classifier, utilizamos os mesmos modelos do AdaBoost e do Bagging: KNN, SVM, Árvore de decisão e Redes neurais. O Voting apresentou desempenho um pouco inferior ao Bagging e superior ao AdaBoost para as divesas configurações, conforme mostrado a seguir.

```
ensemble = VotingClassifier(estimators)
results = model_selection.cross_val_score(ensemble, normalizedX_trainp, y_train_p, cv=kfold)

print(results.mean())
```

```
0.5932147562582345
```

Figura 33: Médias para Voting Classifier

4.6. 3 Stacking

Para o Stacking, foram utilizados os mesmos modelos do Voting. Tanto os modelos base, quanto os utilizados para comparação, foram iniciados com os melhores hiperparâmetros selecionados com a HalvingRandomSearchCV() anteriormente.

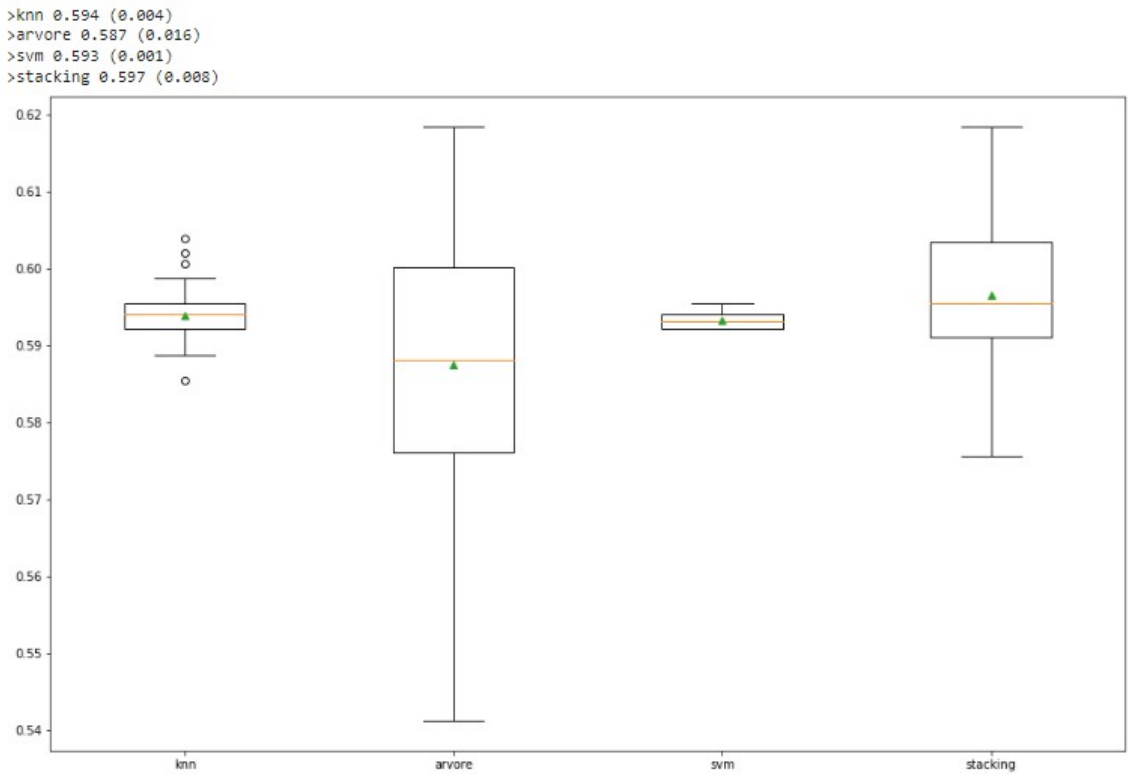

Figura 34: Comparação do desempenho do Stacking

Pelo observado no na figura 35, o Stacking apresentou o melhor desempenho em relação à acurácia (0.597) entre os comitês, e perde para os modelos individuais apenas no caso da Árvore de Decisão (0.61). Sendo assim, a Árvore de Decisão é, pela avaliação das acurácias, o modelo de melhor performance para a base de dados. A decisão final será feita pela avaliação dos modelos a partir da Matriz de Confusão gerada através da aplicação do Confusion Matrix e métricas calculadas pela aplicação do método Classification Report, ambos metrics da biblioteca SciKit Learn.

5. Avaliação dos Modelos e dos resultados

5.1. Análise dos resultados dos desempenhos individuais dos modelos e Teste de Kruskal

Inicialmente, fez-se a avaliação dos desempenhos individuais dos modelos, a partir dos melhores valores de hiperparâmetros obtidos anteriormente por meio da buscas com a `HalvingRandomizedSearchCV()` para cada classificador (KNN, Árvore de Decisão, SVM, Redes neurais e LVQ). Com base na média da aplicação da validação cruzada para testar amostras estratificadas dos dados de treinamento, constata-se que:

```
Desempenhos medios dos modelos:
KNN: 0.588105 (0.029470)
Arvore: 0.548947 (0.035482)
SVM: 0.588421 (0.029342)
Redes Neurais: 0.588421 (0.034816)
LVQ: 0.588421 (0.034287)
```

Figura 35: Desempenhos médios dos modelos

- os desempenhos foram muito semelhantes, no entanto, o classificador que apresentou uma leve melhora na performance foi o KNN (0.588105);
- com algumas repetições da execução, as variações são bastante sutis e, também, a ordem de melhor desempenho pode alterar quando executamos novamente o código.

Em seguida, para comparar os modelos entre si em pares de 2, foi utilizado o teste de Kruskal-Wallis. Ele é indicado para testar a hipótese de que três ou mais populações têm distribuição igual ou não. Para isso, o teste considera que para cada grupo, os indivíduos são diferentes e independentes. Sendo assim, a hipótese nula (H0) adotada e testada por meio de Kruskal-Wallis foi: H0 = Mesma distribuição.

```
Diferentes distribuições (rejeitar H0)

Comparison stats 40.806712637536315
Comparacao KNN | LVQ -> KruskalResult(statistic=25.95587241032802, pvalue=3.493112337813147e-07)
Comparacao KNN | Árvore -> KruskalResult(statistic=0.01908537209850155, pvalue=0.8901220144653048)
Comparacao KNN | SVM -> KruskalResult(statistic=0.04443467404874154, pvalue=0.833046977424799)
Comparacao KNN | Rede Neural -> KruskalResult(statistic=0.028653623607727656, pvalue=0.8655812732973842)
Comparacao LVQ | Árvore -> KruskalResult(statistic=27.0967913389381, pvalue=1.9351887980819832e-07)
Comparacao LVQ | SVM -> KruskalResult(statistic=24.429669616909603, pvalue=7.707254433589925e-07)
Comparacao LVQ | Rede Neural -> KruskalResult(statistic=23.94395674583363, pvalue=9.918113132619187e-07)
Comparacao Árvore | SVM -> KruskalResult(statistic=0.0011922180674952686, pvalue=0.9724557074588055)
Comparacao Árvore | Rede Neural -> KruskalResult(statistic=0.0011916727353427968, pvalue=0.9724620051906415)
Comparacao SVM | Rede Neural -> KruskalResult(statistic=0.0011925052849389697, pvalue=0.9724523911271481)
```

Figura 36: Resultados do teste de Kruskal-Wallis

Ora, os resultados dos rankings do teste de Kruskal da figura 36 mostram que o p-value para todas as comparações apresenta um valor superior ao alpha = 0.05, indício de que as diferenças entre algumas das medianas não são estatisticamente significativas. Assim, os resultados da aplicação dos classificadores nas amostras possuem distribuições diferentes (como já foi observado), e disto resulta a rejeição de H0.

5.2 Resultados da validação cruzada e conclusão da avaliação dos modelos

Se forem observadas somente as acurácias resultantes da predição dos dados de validação com base nos de treinamento parcial nas figuras a seguir, não será possível chegar a um veredicto sobre qual classificador apresentou melhor desempenho. Isto se deve às acurácias possuírem pouca variação entre si – a apresentação com apenas duas casas decimais dificulta ainda mais a análise dessa métrica.

No entanto, outras métricas podem ser analisadas, tais como o True Positive Rate (recall ou sensibility), proporção de positivos verdadeiros do total de positivos, e a Precision (precisão), proporção de positivos verdadeiros do total dos exemplos classificados como positivos.

5.2.1 KNN

```
Acuracia KNN: Treinamento 0.5922023182297155  Teste 0.5873684210526315
Clasification report:
              precision    recall  f1-score   support

           0       0.40      0.03      0.05       320
           1       0.59      0.98      0.74       559
           2       0.00      0.00      0.00        71

    accuracy                           0.59       950
   macro avg       0.33      0.34      0.26       950
weighted avg       0.48      0.59      0.45       950

Confussion matrix:
 [[  8 312   0]
 [  9 550   0]
 [  3  68   0]]
```

Figura 37: Métricas de avaliação do KNN

No KNN é possível observar uma precisão de 40% para a classe 0 (desconhecido), 59% para a classe 1 (masculino) e 0% para a classe 2 (feminino) e uma sensibilidade de 3% para a classe 0 (desconhecido), 98% para a classe 1 (masculino) e 0% para a classe 2 (feminino). Ou seja, embora o KNN tenha apresentado um desempenho bastante satisfatório nas etapas anteriores, ele não conseguiu classificar bem os resultados, uma vez que houve imprecisão na classificação dos dados de acordo com a classe 2 (feminino).

Esta análise é reforçada pela matriz de confusão, uma vez que ela apresenta (mais especificamente) as quantidades de classificações corretas e incorretas, distribuídas entre as classes, como pode ser observado na Figura 40. Na matriz de confusão do KNN, observa-se 0 verdadeiros positivos para a classe 2 (figura 39).

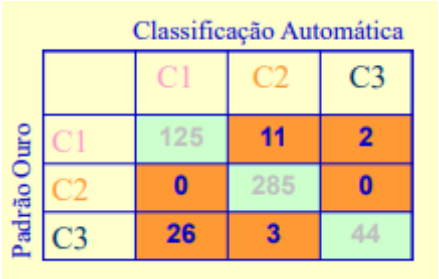

Figura 38: Exemplo de matriz de confusão rotulada para 3 classes ()

5.2.2 LVQ

```
Acuracia LVQ: Treinamento 0.588 Teste 0.588
Clasification report:
              precision    recall  f1-score   support

           0       0.30      0.03      0.05       320
           1       0.59      0.96      0.73       559
           2       0.31      0.07      0.11        71

    accuracy                           0.58       950
   macro avg       0.40      0.35      0.30       950
weighted avg       0.47      0.58      0.46       950

Confussion matrix:
 [[  0 320   0]
 [  0 559   0]
 [  0  71   0]]
```

Figura 39: Métricas de avaliação do LVQ

No LVQ é apresentada uma precisão de 30% para a classe 0 (desconhecido), 59% para a classe 1 (masculino) e 31% para a classe 2 (feminino) e uma sensibilidade de 3% para a classe 0 (desconhecido), 96% para a classe 1 (masculino) e 7% para a classe 2 (feminino). Ou seja, em relação ao KNN, o LVQ apresentaria melhor previsão de verdadeiros positivos para a classe 2, apesar de ter proporção levemente inferior para a classe 0 e similar para a classe 1. No entanto, a matriz de confusão é inconsistente com estes resultados, dado que não há verdadeiros positivos para as classes 0 e 2.

5.2.3 Árvore de Decisão

```
Acuracia Árvore: Treinamento 0.6172286617492097  Teste 0.5789473684210527
Clasification report:
              precision    recall  f1-score   support

           0       0.30      0.03      0.05       320
           1       0.59      0.96      0.73       559
           2       0.31      0.07      0.11        71

    accuracy                           0.58       950
   macro avg       0.40      0.35      0.30       950
weighted avg       0.47      0.58      0.46       950

Confussion matrix:
 [[  8 308   4]
 [ 15 537   7]
 [  4  62   5]]
```

Figura 40: Métricas de avaliação da Árvore de Decisão

Na árvore de decisão é possível observar uma precisão de 30% para a classe 0 (desconhecido), 59% para a classe 1 (masculino) e 31% para a classe 2 (feminino) e uma sensibilidade de 3% para a classe 0 (desconhecido), 96% para a classe 1 (masculino) e 7% para a classe 2 (feminino). Exatamente iguais aos resultados destas métricas para o LVQ.

Embora tenha ocorrido esta coincidência, é possível observar diferenças em outros resultados em relação ao LVQ, como a matriz de confusão, por exemplo. Apesar de as proporções calculadas pelas métricas coincidirem, os números absolutos na matriz de confusão para a árvore de decissão indicam desempenho melhor desse classificador para as classes 0 e 2. A árvore também tem precisão superior à do KNN para a classe 2. É o desempenho mais satisfário em comparação aos modelos anteriores quando se considera as 3 classes.

5.2.4 SVM

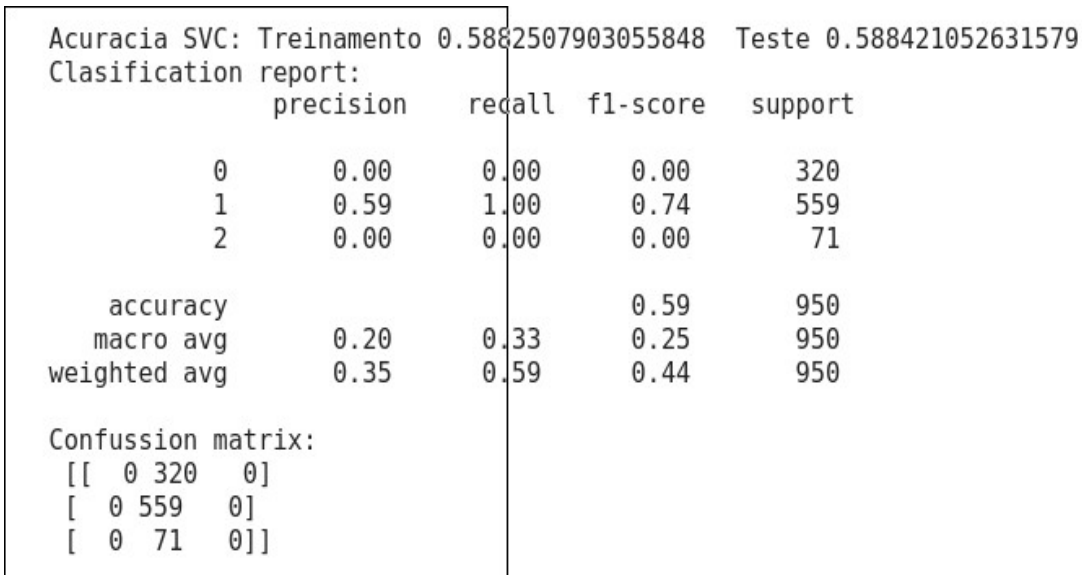

Figura 41: Métricas de avaliação do SVM

O SVM apresentou uma precisão de 0% para a classe 0 (desconhecido), 59% para a classe 1 (masculino) e 0% para a classe 2 (feminino) e uma sensibilidade de 0% para a classe 0 (desconhecido), 100% para a classe 1 (masculino) e 0% para a classe 2 (feminino). Este modelo apresenta pior desempenho que os anteriores tanto para a classe 0 quanto para a classe 2, o que é evidenciado pela matriz de confusão, que não apresenta verdadeiros positivos para elas. **Não se considera que o modelo se adequa à tarefa de classificação**.

5.2.5 Rede Neural

```
Acuracia Rede Neural: Treinamento 0.5882507903055848  Teste 0.588421052631579
Clasification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       320
           1       0.59      1.00      0.74       559
           2       0.00      0.00      0.00        71

    accuracy                           0.59       950
   macro avg       0.20      0.33      0.25       950
weighted avg       0.35      0.59      0.44       950

Confussion matrix:
 [[  0 320   0]
 [  0 559   0]
 [  0  71   0]]
```

Figura 42: Métricas de avaliação do Redes Neurais

Na rede neural é possível observar uma precisão de 0% para a classe 0 (desconhecido), 59% para a classe 1 (masculino) e 0% para a classe 2 (feminino) e uma sensibilidade de 0% para a classe 0 (desconhecido), 100% para a classe 1 (masculino) e 0% para a classe 2 (feminino), bastante semelhante aos resultados destas métricas para o SVM.

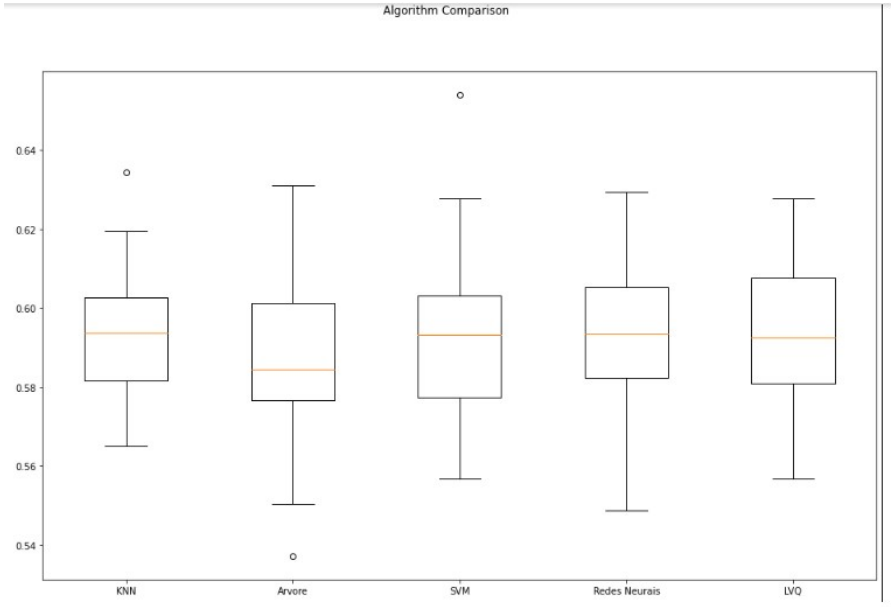

Figura 43: Comparação dos modelos

Pela figura 45, é possível observar que as medianas dos modelos são bastante próximas, exceto para a Árvore de Decisão, que está mais próxima a 0.58 que a 0.60.

5.2.6 Stacking

Para a aplicação do Stacking aos dados de teste, primeiramente criamos um dicionário com os classificadores que integrarão o comitê e um classificador que servirá de meta-aprendiz:

```
# define the base models
level0 = list()
level0.append(('knn', neighbors.KNeighborsClassifier(n_neighbors=98, p=1)))
level0.append(('arvore', DecisionTreeClassifier(criterion='gini', max_depth=6, min_samples_leaf= 2, min_samples_split= 5, random_state=seed)))
level0.append(('svm', SVC(kernel= 'sigmoid', gamma= 0.01, C= 0.1, random_state=seed)))

# define meta learner model
level1 = MLPClassifier(activation= 'logistic', hidden_layer_sizes= 60, random_state=seed)
```

Em seguida, iniciamos o comitê, passando o dicionário como parâmetro e o treinamos para prever os dados do conjunto de dados de teste:

```
# define the stacking ensemble
stacking_test = StackingClassifier(estimators=level0, final_estimator=level1, cv=5)

stacking_test.fit(normalizedX_train, y_train).score(normalizedX_test, y_test)
Y_test_prediction_stacking = stacking_test.predict(normalizedX_test)
```

Por fim, apresentou-se os resultados da classificação com as métricas de avaliação da predição:

```
Acuracia Stacking: Treinamento 0.6111696522655427  Teste 0.5789473684210527
Clasification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       320
           1       0.59      0.97      0.73       559
           2       0.31      0.07      0.11        71

    accuracy                           0.58       950
   macro avg       0.30      0.35      0.28       950
weighted avg       0.37      0.58      0.44       950

Confussion matrix:
 [[  0 316   4]
 [  7 545   7]
 [  1  65   5]]
```

Figura 44*:* Matriz de confusão e métricas de classificação para Stacking

O stacking obteve 59% de precisão na predição da classe 1 (masculino) com 97% de sensibilidade, já para a classe 2 (feminina), apresentou precisão de 31% e sensibilidade de 7%. A classe 0 (desconhecido), apresentou 0% para ambas as métricas, o que indica que o comitê fez o que se esperava: classificou os gêneros desconhecidos de acordo com os outros gêneros, com base nos demais atributos do dataset.

Um valor baixo na precisão e sensibilidade (recall) da classificação da classe 0 (gênero desconhecido), para este caso, é aceitável, pois indica que os usuários/editores inicialmente definidos como desconhecidos foram classificados de acordo com seu gênero mais provável.

5.2.7 ADABoosting

```
Clasification report:
              precision    recall  f1-score   support

           0       0.38      0.26      0.31       320
           1       0.61      0.77      0.68       559
           2       0.50      0.14      0.22        71

    accuracy                           0.55       950
   macro avg       0.50      0.39      0.40       950
weighted avg       0.52      0.55      0.52       950

Confussion matrix:
 [[ 83 231   6]
 [122 433   4]
 [ 15  46  10]]
```

Figura 45: Matriz de confusão e métricas para AdaBoosting com árvore de decisão

```
Clasification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       320
           1       0.59      1.00      0.74       559
           2       0.00      0.00      0.00        71

    accuracy                           0.59       950
   macro avg       0.20      0.33      0.25       950
weighted avg       0.35      0.59      0.44       950

Confussion matrix:
 [[  0 320   0]
 [  0 559   0]
 [  0  71   0]]
```

Figura 46: Matriz de confusão e métricas para AdaBoosting com SVM

Os resultados da formação de um comitê pelo **AdaBoosting com a Árvore de Decisão como classsificador base obtiveram resultados superiores a todos os outros modelos analisados anteriormente**. Houve um número de verdadeiros positivos superior ao de todos os classificadores anteriores e as porcentagens de precisão também se destacam em relação aos resultados vistos anteriormente. O mesmo não ocorre para o comitê que tem o SVM como classificador base, o qual não consegue obter nenhum verdadeiro positivo para as classes 0 e 2 (Desconhecidos e gênero Feminino).

5.2.8 Bagging

```
Árvore de Decisão
Clasification report:
              precision    recall  f1-score   support

           0       0.37      0.06      0.10       320
           1       0.60      0.95      0.74       559
           2       0.50      0.10      0.16        71

    accuracy                           0.59       950
   macro avg       0.49      0.37      0.33       950
weighted avg       0.52      0.59      0.48       950

Confussion matrix:
 [[ 19 299   2]
 [ 23 531   5]
 [  9  55   7]]
SVM
Clasification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       320
           1       0.59      1.00      0.74       559
           2       0.00      0.00      0.00        71

    accuracy                           0.59       950
   macro avg       0.20      0.33      0.25       950
weighted avg       0.35      0.59      0.44       950

Confussion matrix:
 [[  0 320   0]
 [  0 559   0]
 [  0  71   0]]
KNN
Clasification report:
              precision    recall  f1-score   support

           0       0.45      0.02      0.03       320
           1       0.59      0.99      0.74       559
           2       0.00      0.00      0.00        71

    accuracy                           0.59       950
   macro avg       0.35      0.34      0.26       950
weighted avg       0.50      0.59      0.45       950

Confussion matrix:
 [[  5 315   0]
 [  4 555   0]
 [  2  69   0]]
Rede Neural
```

Figura 47: Matriz de confusão e métricas para cada classificador-base

```
Rede Neural
Clasification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       320
           1       0.59      1.00      0.74       559
           2       0.00      0.00      0.00        71

    accuracy                           0.59       950
   macro avg       0.20      0.33      0.25       950
weighted avg       0.35      0.59      0.44       950

Confussion matrix:
 [[  0 320   0]
 [  0 559   0]
 [  0  71   0]]
```

Figura 48: Matriz de confusão e métricas para classificador-base Rede Neural

A maior parte dos resultados da aplicação do “Bagging” para os diversos classificadores-base também se mostram menos aceitáveis que os do Adaboosting com a Árvore de Decisão. Tanto para SVM, como para KNN e Rede Neural, o número de verdadeiros positivos para as classes 0 e 2 é pequeno ou 0, bem como as métricas de precisão e sensibilidade. O baggin com Árvore de decisão teve resultado bastante similar ao do AdaBoosting com a árvore em termos de precisão, mas

5.2.9 Voting

```
Clasification report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       320
           1       0.59      1.00      0.74       559
           2       0.00      0.00      0.00        71

    accuracy                           0.59       950
   macro avg       0.20      0.33      0.25       950
weighted avg       0.35      0.59      0.44       950

Confussion matrix:
 [[  0 320   0]
 [  1 558   0]
 [  0  71   0]]
```

Figura 49: Matriz de confusão e métricas para comitê Voting

As métricas resultantes da aplicação do Voting não são muito interessantes para a resolução do problema. Como observado outros dos testes realizados, há pouca precisão para as classes 0 e 2, assim como sensibilidade 0 para estas. Este comitê não está entre os mais adequados para ser aplicado ao problema de classificação proposto.

6. Conclusões e Trabalhos Futuros

É notória a dificuldade dos modelos em predizer membros das classes 0 e 2. A tendência é esperada, dado que a classe 2 (gênero feminino) é subrepresentada na plataforma Wikipedia e, por conseguinte, no conjunto de dados utilizado e a classe 0 (gênero desconhecido), apesar de numerosax, não possui comportamento de padrão facilmente identificável, por incluir tanto pessoas do gênero feminino como masculino e outros gêneros não possíveis de auto-declaração pelos formulários da plataforma.

Observa-se desempenho satisfatório da Árvore de Decisão como classificador individual e resultados ainda melhores quando este modelo é aplicado como classificador-base em comitês de classificação. Pelos experimentos e análises realizados, indica-se o **Bagging com Árvore de Decisão como base** como classificador mais adequado para prever as classes do problema de definição do gênero de editores da Wikipedia em língua espanhola, dado o seu desempenho satisfatório de predições acuradas e erro diminuto em relação a todos os modelos e arranjos testados nesta análise.

Referências

COLLIER, Benjamin; BEAR, Julia. **Conflict, criticism, or confidence**: an empirical examination of the gender gap in wikipedia contributions. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (*CSCW '12*). Association for Computing Machinery, New York, NY, 2012. P. 383–392. https://doi.org/10.1145/2145204.2145265

COHEN, Noam. "Define gender gap? Look up Wikipedia's contributor list." *The New York Times* 30.01 (2011)

CASSEL, J. **Editing wars behind the scenes**. New York Times, Publicado em Fevereio de 2011. Disponível em: https://www.nytimes.com/roomfordebate/2011/02/02/where-are-the-women-in wikipedia/a-culture-of-editing-wars Acesso em: 21 de maio de 2022.

GLOTT, Ruediger; GHOSH, Rishab. Analysis of Wikipedia Survey Data. **wikipediastudy. org, March**, 2010.

GRAHAM, Mark; STRAUMANN, Ralph K.; HOGAN, Bernie. Digital divisions of labor and informational magnetism: Mapping participation in Wikipedia. **Annals of the Association of American Geographers**, v. 105, n. 6, p. 1158-1178, 2015.

ENCYCLOPEDIA BRITANNICA. **"Encyclopédie".** 2022. Disponível em: https://www.britannica.com/topic/Encyclopedie. Acceso em: 21 de maio de 2022.

HILL, BM, SHAW, A. **The Wikipedia Gender Gap Revisited**: Characterizing Survey Response Bias with Propensity Score Estimation. 2013. PLoS ONE 8(6): e65782. doi:10.1371/journal.pone.0065782

LUNA, Z. **CRISP-DM Fase 1:** Entendimento de Negócios. Medium. 2021. Disponível em: https://medium.com/analytics-vidhya/crisp-dm-phase-1-business-understanding-255b47adf90a. Acesso em: 21 mai. 2022.

MINGUILLÓN, J.; MENESES, J.; AIBAR, E. FERRAN-FERRER, N. FÀBREGUES, S. Exploring the gender gap in the Spanish Wikipedia: Differences in engagement and editing practices. **PLoS ONE**, vol. 16, n. 2, 2021. Disponível em: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0246702. Acesso em: 21 mai. 2022.

UCI Machine Learning. **Gender Gap in Spanish WP Data Set.** Disponível em: https://archive.ics.uci.edu/ml/datasets/Gender+Gap+in+Spanish+WP#. Acesso em: 21 mai. 2022.

WIKIPEDIA. **Wikipedia Editing Policy**. Disponível em: https://en.wikipedia.org/wiki/Wikipedia:Editing_policy#Be_cautious_with_major_changes:_discuss Acesso em: 21 de maio de 2022.

WIKIMEDIA FOUNDATION. **Estatísticas**. Disponível em: https://stats.wikimedia.org/#/all-projects Acesso em: 21 de maio de 2022.

GLOTT http://www.wikipediastudy.org/docs/Wikipedia_Age_Gender_30March%202010-FINAL-3.pdf

MINITAB 19. Interpretar os principais resultados para Teste de Kruskal-Wallis. Disponível em: https://support.minitab.com/pt-br/minitab/19/help-and-how-to/statistics/nonparametrics/how-to/kruskal-wallis-test/interpret-the-results/key-results/. Acesso em: 25 jun. 2022.

ANEXO 1 – Árvore de Decisão com os dados Base

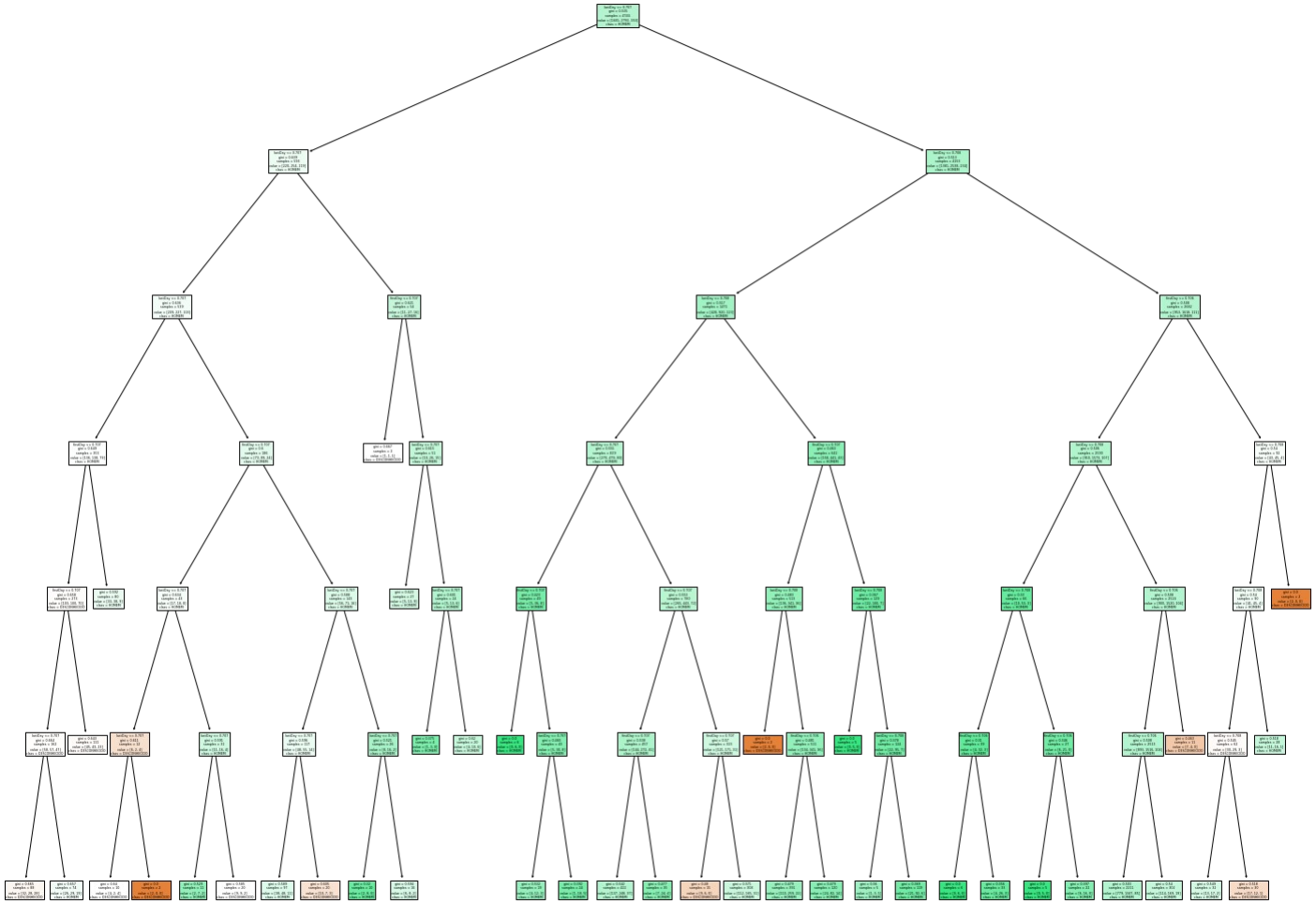

Figura 50: Árvore para o dataset base

ANEXO 1 – Árvore de Decisão após seleção de atributos

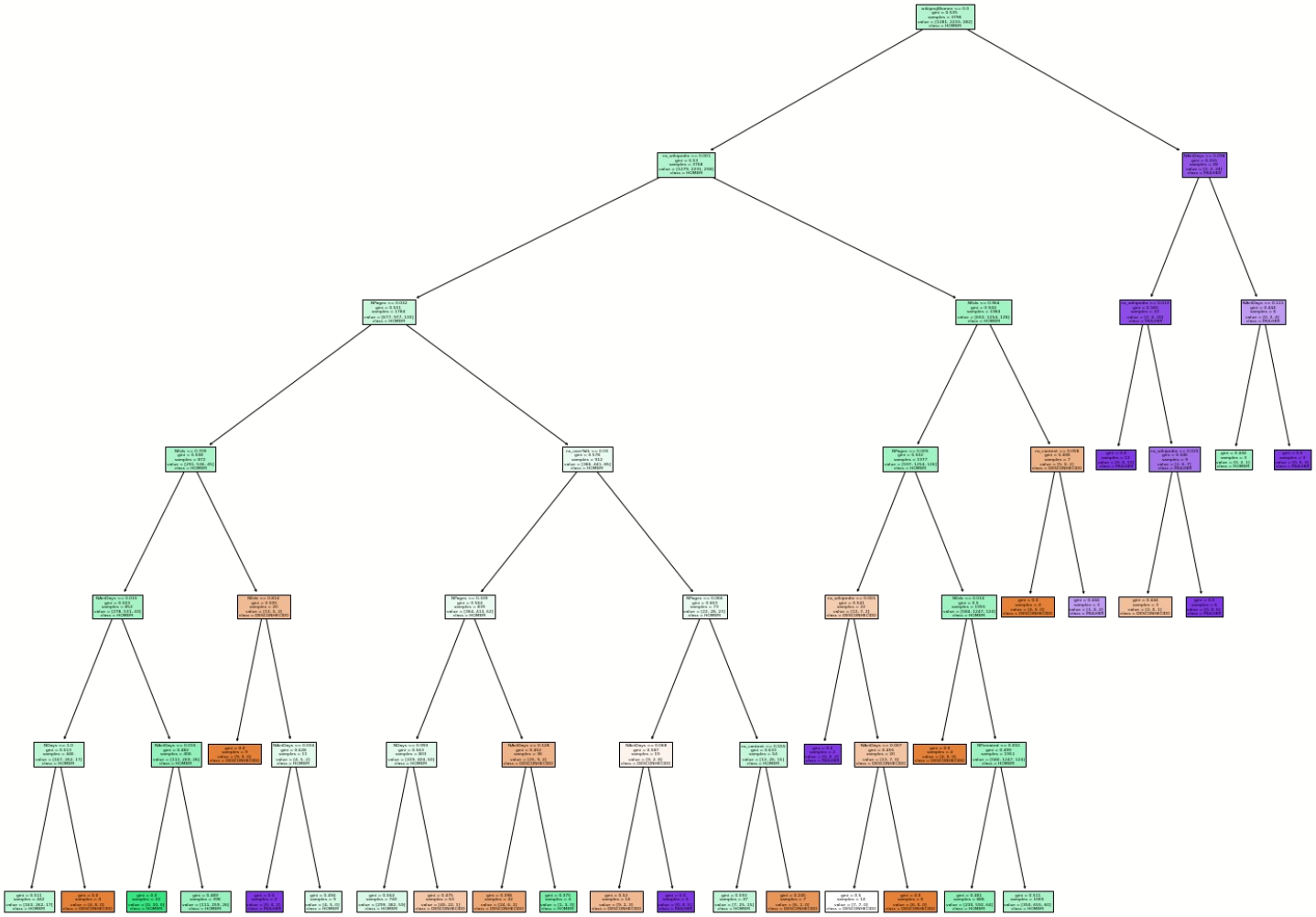

Figura 51: Árvore de decisão para o conjunto de treinamento