Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital
Bacharelado em Tecnologia da Informação

Disciplina: Aprendizado de Máquina
Discentes:
Gabriel Cristian Melo da Silva
Gabriel Estácio de Souza Passos
Rodrigo Faustino de Sousa

Docente: SAMUEL DA SILVA OLIVEIRA

IMD1101-Aprendizado de Máquina - 2020.2 Check-Point Modelos Supervisionados

1. **Objetivos**:

O principal objetivo deste check-point é analisar como os métodos e técnicas de aprendizado supervisionado se comportam em uma aplicação prática.
Para tal, o aluno deverá comparar os resultados obtidos com os algoritmos vistos em sala de aula (Redes Neurais, k-NN, Árvore de Decisão e Naive Bayes) quando aplicados as bases de dados escolhidas por cada grupo (uma base de dados original e 3 bases reduzidas). Para a realização desta comparação, alguns objetivos específicos são desejados, que são os seguintes:

I. Verificar o impacto do parâmetro *k* no k-NN, além de analisar o impacto do uso ou não do processo de escalonamento dos valores dos atributos numéricos para [0,1] e do uso de pesos no erro de classificação do k-NN.
II. Verificar como o erro de classificação e o tamanho da árvore de decisão dependem do parâmetro de poda (poda ou sem poda).
III. Verificar o impacto das duas suposições principais do Naive Bayes, os dados obedecem uma distribuição normal? Os dados possuem alguma relação?
IV. Verificar como o erro de classificação depende do tamanho da Rede Neural, como também da taxa de aprendizado e do número de iterações.

2. Metodologia do trabalho

Como já sabemos, as três principais etapas do processo de Aprendizado de Máquina são as seguintes: análise dos dados, métodos utilizados e pós-processamento.

Na etapa de métodos utilizados, que é o objeto deste check-point, serão analisados os algoritmos supervisionados vistos em sala de aula. Os resultados obtidos devem ser interpretados e apresentados em forma de relatório. Ao final, cada grupo deve preparar um relatório, dedicando bastante tempo à interpretação dos resultados obtidos. Cada grupo deve decidir como serão ilustrados/discutidos os resultados obtidos.

2.1 Métodos de Aprendizado Supervisionados

Como já mencionado, quatro métodos de aprendizado de máquina devem ser utilizados neste trabalho, que são: redes neurais (Multi-LayerPerceptron), k-NN, Árvores de decisão e o Naive Bayes (se você estiver trabalhando com regressão os algoritmos são os mesmos, mas para regressão, com exceção do Naive Bayes que será substituído pelo Random Forest). Para cada método supervisionado, é preciso responder detalhadamente as seguintes perguntas:

- **Qual foi o melhor conjunto de parâmetros para a base de dados escolhida, por que? Em outras palavras, existe alguma análise, a nível de parâmetros e/ou dados utilizados, que possa justificar e/ou explicar o comportamento do método em questão?**

k-NN:

O conjunto de parâmetros que gerou o melhor resultado no k-NN foi:

```
n_neighbors=2, metric='euclidean'
```

onde n_neighbors corresponde ao k, no nosso melhor caso, igual a 2, e metric corresponde ao modelo de distância que utilizamos, no caso, a euclidiana.

Árvores de decisão:

O conjunto de parâmetros que gerou o melhor resultado na árvore de decisão foi:

```
max_depth = 10, min_samples_split=10, min_samples_leaf=10
```

onde max_depth corresponde a profundidade máxima da árvore, min_samples_split corresponde a quantidade mínima de observações de uma classe para que um nó faça o split, e min_samples_leaf corresponde a quantidade mínima de observações de uma classe para que seja criado um nó terminal (folha).

Naive Bayes:

Em questão de parâmetros, não há muito que se falar, o NB foi executado sem uso de parâmetros adicionais. Já na questão dos dados, essa parte é discutida mais a frente em relação à hipótese de normalidade e correlação. Em resumo, mesmo que não tenha sido o caso a comprovação de uma distribuição normal e independência dos dados, o impacto não foi alto no resultado. O NB ele é robusto e respondeu bem a nossa massa de dados.

MLP:

- **Os mesmos conjuntos de parâmetros obtiveram melhor desempenho nas 4 bases de dados que vcs estão avaliando?**

k-NN:

Sim. Por mais que o melhor caso de todos os observados tenha a presença do parâmetro ‘weight’, esse foi o único caso que este parâmetro gerou impacto positivo. Em todos os outros, ou os resultados pioraram, ou não sofreram alteração nenhuma. A presença do weight também resultava na presença de falsos positivos, o que não aconteceu sem ele.

Árvores de decisão:

Sim. Em todas as bases esse conjunto de parâmetros resultou numa diminuição do número de nós, ao mesmo tempo que mantinha ou até aumentava a acurácia do modelo.

Naive Bayes:

Todas as bases foram executadas sem parâmetros adicionais.

MLP:

- **A redução dos dados foi benéfica para a sua base de dados? Se sim, qual foi a melhor redução (em termos de melhora de desempenho)?**

k-NN:

Sim. Apesar de alguns casos onde a base original tem desempenho levemente melhor, as bases reduzidas apresentaram resultados mais consistentes, com menos variações nas métricas. A que apresentou o melhor desempenho foi a BaseReduzida3, resultado da extração de atributos usando PCA.

Árvores de decisão:

Não. A base original foi a que apresentou melhor desempenho, tanto antes da poda, quanto depois.

Naive Bayes:

Sim. A base que melhor respondeu ao método foi a BaseReduzida3, mas tanto a BaseReduzida1 e BaseReduzida2 obtiveram resultados melhor que a base original.

MLP:

Aqui é extremamente importante não se deter APENAS nos resultados e dizer que, por ter dado o menor erro, o conjunto de parâmetro X foi o melhor. Mas sim, o aluno deve tentar entender e justificar o comportamento do método para a base de dados escolhida. Nas próximas subseções, descreve-se melhor os métodos individualmente. Lembrando que todos os testes devem ser feitos 10 vezes, variado a seed do kFold e do algoritmo (para os algoritmos que tiverem esse atributo). No final o valor utilizado como métrica de resultado é o valor médio da acurácia e do desvio padrão.

2.1.1 k-NN:

Para cada base de dados, serão feitos experimentos em dois cenários: uso ou não do escalonamento dos valores dos atributos numéricos para [0,1]. Em cada um desses contextos, o procedimento deve ser o seguinte. Serão feitos treinamentos usando 10-fold cross validation para diferentes valores de *k* (devem ser escolhidos, pelo menos, três valores diferentes de *k*). Aqui, o grupo deve buscar o valor de *k* que produza o melhor resultado para esta base de dados.

Fazer os experimentos com e sem peso (utilizado para desempate, caso haja empate). Após realizar os experimentos com este método, o aluno deve buscar respostas para as seguintes perguntas:

1. Qual foi o melhor valor de *k*?

O melhor valor de k dos três testados (2, 5 e 8) foi 2, pois foi o que apresentou maior média em cada métrica e foi com k = 2 que aconteceu o melhor caso de teste.

2. Foi importante escalonar os valores?

Somente quando o teste era realizado com peso. Quando sem peso, o escalonamento não gerou nenhum impacto prático no resultado.

3. E o peso teve algum impacto no desempenho do método?

Somente no k = 2 e na BaseReduzida3 (base que fizemos extração utilizando PCA). Todos os parâmetros de avaliação melhoraram com o uso de peso no caso citado, porém nos outros casos, a presença do parâmetro ‘weight’ (independente da lógica de aplicação de peso) causava uma leve queda de desempenho.

2.1.2 Árvores de decisão:

Para cada base de dados, serão feitos treinamentos usando 10-fold cross validation,variando se apenas o parâmetro poda (ou seja, construir árvore com poda e sem poda). Aqui, as perguntas a serem respondidas são as seguintes:

1. A árvore (Árvore de Decisão) apresentou overfitting antes da poda?
Sim. A árvore apresentou uma quantidade muito elevada de nós.

2. Com o uso da poda é possível afirmar se a AD estava ou não com overfitting? Sim. Utilizamos a função tree_.node_count para calcular o número de nós da árvore para cada base e fazer os ajustes necessários. Depois disso, através do método de erro reduzido, fizemos a poda até diminuirmos a taxa de erro. Para isso, usamos os parâmetros max_depth, min_samples_split e min_samples_leaf. Isso reduziu significativamente o número de nós, que caiu, aproximadamente, para a metade do número original em todas as quatro bases.

2.1.3 Random Forest:

Para cada base de dados, serão feitos treinamentos usando 10-fold cross validation,variando se o parâmetro número de estimadores e o número de amostras para cada estimador (nesse caso, o parâmetro *boostrap* deve ser True). Deve-se trabalhar com 3 valores diferentes de estimadores e amostras. Os valores podem ser escolhidos de forma arbitrária, mas o número total de amostras deve ser dividido de forma igualitária para cada estimador. Se a floresta possui dois estimadores, então metade das amostras deverá ser usada em cada estimador. Se tenho três estimadores, então cada estimador vai ter um terço das amostras. Para cada quantidade de estimador escolhido, deve ser testado com a quantidade de amostras escolhidas para cada estimador e também ser testado com todas as mostras da bases para cada estimador (parâmetro *boostrap* deve ser False). Aqui, as perguntas a serem respondidas são as seguintes:

1. A alteração na quantidade estimadores e amostras teve efeito positivo ou negativo?

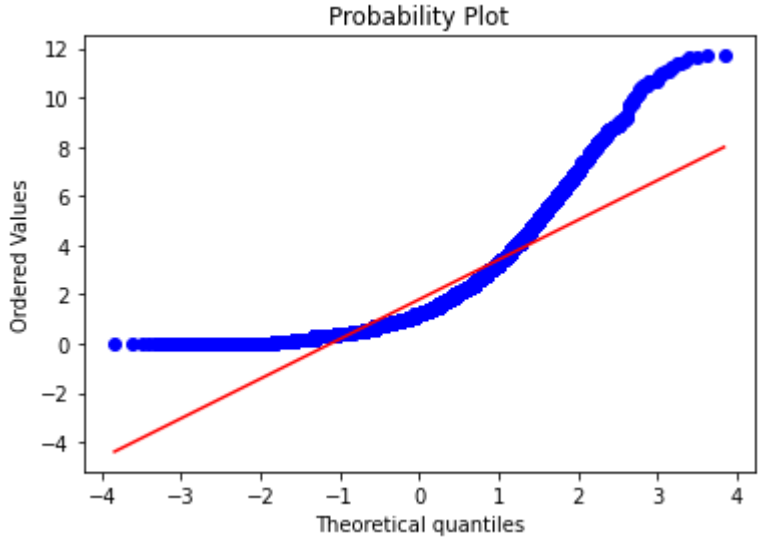
2. A floresta obteve resultado melhor quando usava todas as amostras da basea para os estimadores ou quando os estimadores usavam X quantidades de amostras da base?

2.1.4 Naive Bayes:

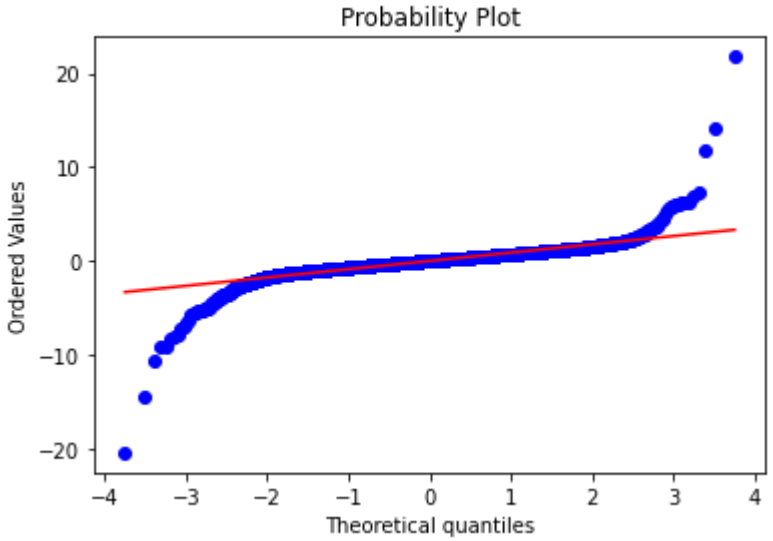
Os experimentos serão feitos usando 10-fold cross validation. Para cada base de dados, as perguntas a serem respondidas para o NB são as seguintes:

1. Os meus dados numéricos estavam obedecendo uma distribuição normal?
Foi feita análise dos Q-Q Plots para se verificar de forma visual se os dados estavam seguindo ou não uma distribuição normal. Foi tentado também utilizar o teste de Shapiro-Wilk, mas como a quantidade de instâncias é muito alta, o p-value do teste não se comporta bem. O que os gráficos sugeriram é que boa parte dos dados numéricos não seguem uma distribuição normal. Houveram casos onde se parecia ter alinhamento dos resultados na reta do Q-Q Plot, mas no rabo e na cabeça a distância para reta aumentava bastante. A diferença entre os gráficos relativos às bases não foi muito

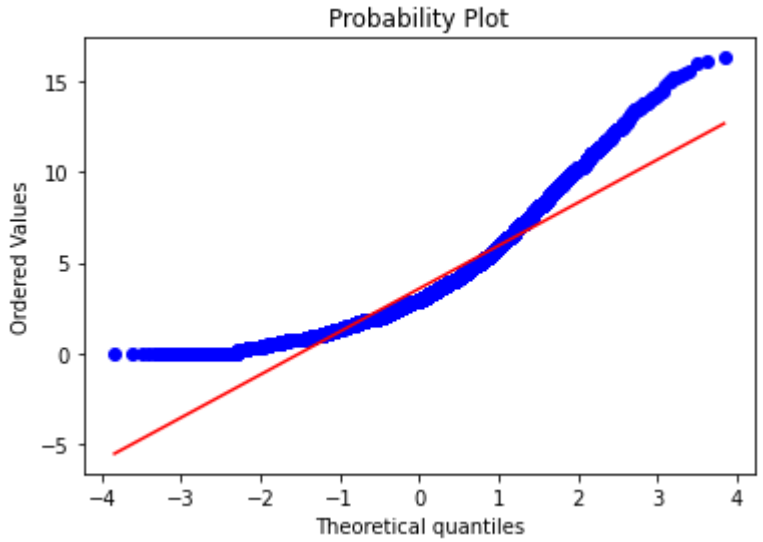
discrepante, as mudanças maiores ocorreram em relação à base reduzida quatro. Os Q-Q Plots da base quatro são os que ficam mais próximos de uma distribuição normal, segue os gráficos para comparação.



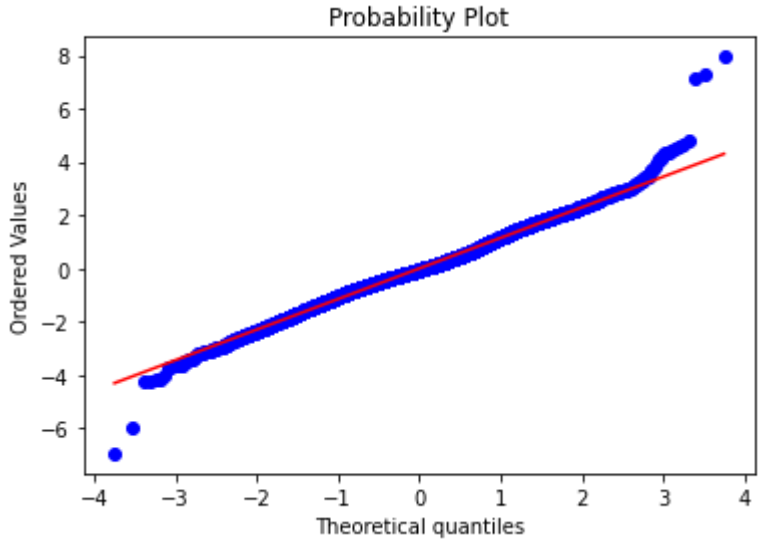
anexo 1- astDB1.png



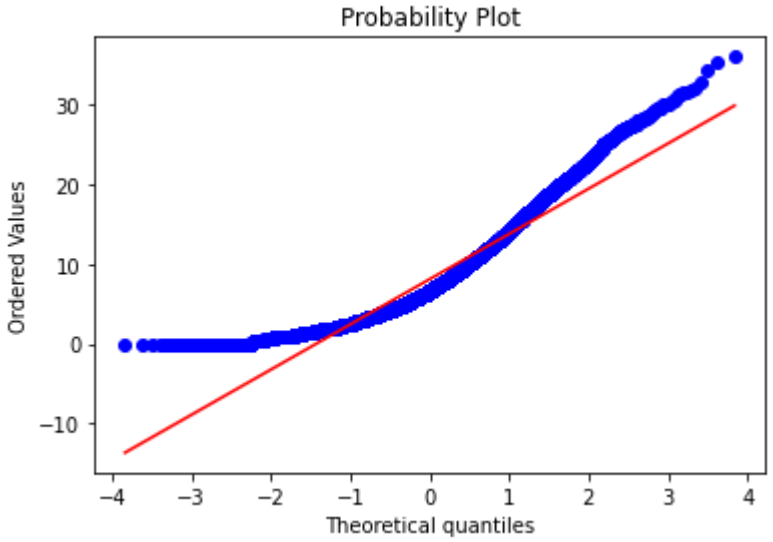
anexo 2 - astDB4.png



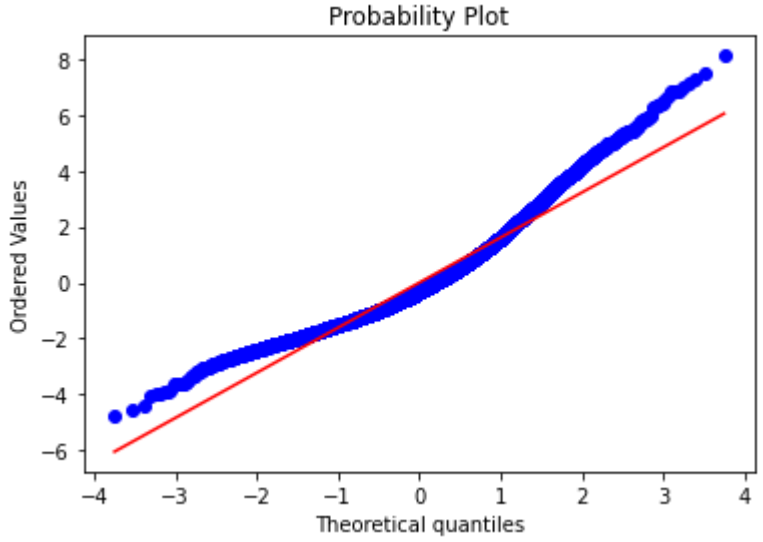
anexo 3 - rebDB4.png



anexo 4 - rebDB4.png



anexo 5 - ptsDB1.png



anexo 6 - ptsDB4.png

2. Os meus dados possuem alguma relação?

Para verificar essa questão, foi gerado uma matriz de correlação linear de Pearson. A correlação mais forte encontrada foi entre os atributos “ast” e

"ast_pct", com valor arredondado de 0,8044. Trago abaixo o resultado tabelado e destacado dos valores que são interessantes:

	player_id	gp	pts	reb	ast	net_rating	oreb_pct	dreb_pct
player_id	1	0,008672387458987746	-0,004713339660675827	-0,021567992493343673	0,002404488817704838	0,006179895808298878	-0,0029317728513230164	-0,0284940349650616
gp	0,008672387458987746	1	0,542958347460514	0,4693768732651626	0,3857567950329448	0,2448225507764485	-0,0186239211285257 75	0,06968276458494044
pts	-0,004713339660675827	0,542958347460514	1	0,6205694078422116	0,6468639759654748	0,21401820309781802	-0,11366026278634025	0,05429184962482412
reb	-0,021567992493343673	0,4693768732651626	0,6205694078422116	1	0,2171235177870716	0,18393929636264006	0,42071729185438717	0,6175481312551714
ast	0,002404488817704838	0,3857567950329448	0,6468639759654748	0,2171235177870716	1	0,1632267489089757	-0,3455024951617714	-0,2133768273626274
net_rating	0,006179895808298878	0,2448225507764485	0,21401820309781802	0,18393929636264006	0,1632267489089757	1	0,06110197649662752	0,0321022846786514
oreb_pct	-0,0029317728513230164	-0,018623921128525775	-0,11366026278634025	0,42071729185438717	-0,3455024951617714	0,06110197649662752	1	0,5544345793020987
dreb_pct	-0,02849403496506166	0,06968276458494044	0,05429184962482412 6	0,6175481312551714	-0,21337682736262747	0,0321022846786514	0,5544345793020987	
usg_pct	0,01882858071515491	0,13499252373471068	0,6355353053484072	0,21905164466543384	0,38067127753311353	-0,02379365630642402	-0,1058624746611285	-0,0358735121152224
ts_pct	-0,008456567602855845	0,3841680575674144	0,3764269506473027	0,30878749376442755	0,17208390548494593	0,24706055753810657	0,06116599830967255	0,1049763599049909
ast_pct	0,015741214414152452	0,13105699847827487	0,316653835175423	-0,09498729593245667	0,8044153398972488	0,05682542077617916 4	-0,44609896228098844	-0,352206244103546
season	0,015169164631059712	-0,05457187894618867	0,023598603920543897	-0,0070489650481829775	0,0026881774911530457	0,008259202715238905	-0,14348869807599315	0,022685594384637 58

anexo 7 - tabela com a matriz de correlação Pearson

No caso, vale ressaltar que as correlações mais relevantes positivas foram entre "gp" x "pts", "pts" x "reb", "pts" x "ast", "pts" x "usg_pct" e "reb" x "dreb_pct". Das negativas, não chegaram no patamar de acima de 0,5, mas é interessante ressaltar que os menores valores negativos foram entre "dreb_pct" x "ast_pct" e "oreb_pct" x "ast_pct". A tabela contendo os resultados para a base quatro ficou com valores muito pequenos, todos fracionários em notação de expoente negativo. Segue imagem:

	pts	reb	ast	net_rating	usg_pct	ast_pct
pts	1.0	-4.624462301172194e-17	2.0966450888426218e-19	-1.5740156907981196e-17	-6.465203309707668e-18	-1.5239030168339412e-17
reb	-4.624462301172194e-17	1.0	-3.348274097071889e-16	-3.345139201547554e-17	-7.291996101267264e-18	8.790483531505267e-17
ast	2.0966450888426218e-19	-3.348274097071889e-16	1.0	-3.539716307308 8556e-16	-2.6372665569846564e-16	2.8409811940092386e-16
net_rating	-1.5740156907981196e-17	-3.345139201547554e-17	-3.539716307308 8556e-16	1.0	8.416908821212572e-17	-2.3316073668933586e-16
usg_pct	-6.465203309707668e-18	-7.291996101267264e-18	-2.6372665569846564e-16	8.416908821212572e-17	1.0	-1.4363859650682696e-16
ast_pct	-1.5239030168339412e-17	8.790483531505267e-17	2.8409811940092386e-16	-2.3316073668933586e-16	-1.4363859650682696e-16	1.0

anexo 8 - tabela base reduzida quatro

2.1.5 Redes Neurais:

Para cada base de dados, serão feitos treinamentos de redes MLP usando o *backpropagation* padrão (com o termo momentum fixo de 0.8), variando-se os seguintes parâmetros: a. Quantidade máxima de iterações (ou ciclos).

b. Quantidade de neurônios intermediários (ou escondidos) da rede --- serão usadas redes com apenas uma camada intermediária.

c. Taxa de aprendizado.

Para cada um destes três parâmetros, devem ser usados três valores. Sendo assim, o grupo deve escolher três valores diferentes para: quantidades máxima de iterações, quantidades de neurônios escondidos e taxa de aprendizado. Exemplo: poderiam ser usados 100, 1.000 e 10.000 iterações; 4, 8 e 12 neurônios escondidos; e taxas de aprendizado de 0.1, 0.01 e 0.001. Isto é apenas um exemplo.

O próprio grupo é quem vai escolher os valores a serem usados, dependendo do problema de classificação a ser resolvido. Vale a pena lembrar que não existem valores ideais que podem ser usados para qualquer problema, de modo que uma taxa de aprendizado de, por exemplo, 0.01 pode ser "pequena demais" para um determinado problema, sendo "grande demais" para outro. O mesmo é verdade para a quantidade de iterações e de nodos escondidos. Porém, para a taxa de aprendizado, há uma particularidade que o aluno deve obedecer.

O grupo não precisa escolher exatamente os valores supracitados, mas é necessário que o grupo escolha valores com granularidades diferentes. Portanto, deve-se escolher os seguintes valores 0.V; 0.0V e 0.00V, onde V pode ser qualquer valor inteiro no intervalo [1,9]. Para os outros dois parâmetros, é importantes que não seja escolhidos valores muitos próximos (por exemplo: 5, 7 e 9 neurônios na camada escondida), para que tenhamos um quadro mais genérico do comportamento da rede quando variando estes parâmetros.

Uma sugestão para o número de neurônios seria: usar o valor (num. Att + num. classes)/2 como ponto de partida. Para o numero de iterações, a minha sugestão seria 100, 1.000 e 10.000,

como sugerida acima.

Para a execução do treinamento, para cada combinação dos três parâmetros supracitados, deve-se usar um método 2-fold-cross-validation, sendo anotados os respectivos resultados. Portanto, devem ser executados 3 x 3 x 3 = 27 treinamentos preliminares. Uma vez feita as 27 execuções, deve ser escolhida a melhor rede obtida. Em outras palavras, o grupo irá escolher a melhor combinação de valores para estes três parâmetros. Define-se como "melhor rede" a rede que proporcionou os melhores resultados, sendo o mais compacta possível. Uma boa escolha pode ser baseada no erro (ou precisão) do conjunto de treinamento e no tamanho da rede. Exemplo: se o menor erro de treinamento foi obtido usando 1.000 iterações, oito nodos escondidos e taxa de aprendizado de 0.001, então este pode ser considerado o melhor conjunto de parâmetros (ou seja, a melhor rede). Caso haja um empate nas precisões de mais de uma rede, dê prioridade as redes mais simples. Para o conjunto de parâmetros escolhido (melhor rede), deve ser aplicado o 10-fold cross validation.

Aqui, o aluno deve tentar responder as seguintes perguntas:

1. Qual intervalo de valores a rede estava com underfitting ou com overfitting?

O modelo não sofreu overfitting de modo que os parâmetros não tornam o modelo complexo o suficiente para que ele se adeque por completo à base de dados. Já o underfitting ocorreu em todos os testes feitos com o modelo de hidden_layer_sizes = 20 e learning_rate_init = 0.1. Neste caso, o modelo performou em média 20% abaixo do modelo com melhor resultado.

2. O que acontece quando eu fixo os dois parâmetros e vario a taxa de aprendizado?

Houve uma diminuição da acurácia e um aumento da perda do modelo. Isto permite que o modelo atinja uma performance boa mais rápido ao custo de uma maior inconsistência durante o treino.

3. E se o aluno fixar dois parâmetros e variar o numero de neurônios na camada escondida, o que acontece? A mesma pergunta para o numero de iterações?

O aumento de neurônios deixa o modelo mais complexo. Assim, capaz de reconhecer padrões mais abstratos ao custo de um maior risco de overfitting e um maior custo computacional no treinamento. Já a diminuição do número de neurônios permite uma menor necessidade computacional durante o treino ao custo do risco de underfitting.

Já com o número de iterações, o modelo deve ser constantemente avaliado durante o treino, ao aumentar os valores, a acurácia sobre as instâncias de treino também aumentava, tendendo ao overfitting.

O melhor resultado, levando em consideração o desvio padrão, foi obtido a partir da base original.

2.1.6 Análise Comparativa:

Neste caso, é importante fazer duas análises:

1) Nesta primeira análise, o aluno deve responder se a redução de dados teve impacto positivo ou negativo no desempenho dos modelos supervisionados. Para responder a pergunta, monte a seguinte tabela:

	k-NN	AD	NB	NN	RF

Base original	99,82%	99,82%	97,7%	93,4%	Acurácia média
Base reduzida 1	99,74%	99,74%	98,0%	92,28%	Acurácia média
Base reduzida 2	99,74%	99,74%	98,0%	92,54%	Acurácia média
Base reduzida 3	99,74%	99,74%	99,1%	92,45%	Acurácia média
Média geral	99,76%	99,76%	98,2%	92,80%	Acc Média gera

2) Nesta segunda análise, o aluno deve responder qual modelo obteve o melhor desempenho na sua base dados. Para tal, analise a tabela descrita na tabela anterior e escolha a base de dados com a melhor accurácia média e construa a seguinte tabela:

	k-NN	AD	NB	NN	RF
Melhor base	Base Original (0,9982±0,059)	Base Original (0,9982±0,051)	Base reduzida 3 (0,9820±0,072)	Base Original(0,9345±0,056)	Acc±DP

Onde:

Acc: é a mehor acurácia de todos os parâmetros analisados (quem está trabalhando com regressão ao invés da acurácia usar a medida de erro MAPE);

DP: é o desvio padrão. Para calcular o DP, é preciso pegar as melhores configurações e aplique o 3x10-fold CV, com seeds diferentes. Daí calcule o desvio padrão destes 30 resultados.