Faculdade de Tecnologia
Universidade Estadual de Campinas

CET0621 – Aprendizado de Máquina na Análise de Dados
Lista de Exercícios 2

Nome do Aluno 1: Gustavo Ferreira Lima **RA: 2023611300**

Nome do Aluno 2: Mateus de Almeida Frigo **RA: 2023611431**

Instruções

- Esta lista de exercícios deve ser desenvolvida em **trios**;
- As respostas devem estar em um arquivo PDF, juntamente com o **nome e RA dos membros do trio**;
- **Apenas um dos membros** deve postar o arquivo no Moodle;
- Esta lista contém questões teóricas e práticas.

Questões:

1) (3,0pt) Em uma clínica médica, especializada em uma determinada doença infecciosa, existe uma base de dados de pacientes que armazena informações referentes a **quatro atributos**: idade do paciente (valor numérico); nível de duas substâncias A e B no sangue (valores numéricos); e presença ou não de histórico da doença na família (valor binário). Cada amostra desta base pode ser de um paciente doente ou não (classes do problema), e tal informação também é dada. Diante deste cenário, pede-se:

   a) Considerando-se a utilização de uma rede neural do tipo **Perceptron Multicamadas** (MLP) como classificador de dados, supondo que sejam adotados **10 (dez) neurônios** na **única camada oculta** da rede e que **todos os neurônios tenham *bias***, indique o **número total de entradas** e **de saídas** que você definiria para esta rede e calcule o **número total de pesos** a serem ajustados durante a etapa de treinamento.

   b) Ao treinar a MLP indicada no item (a) você observou que, apesar da rede apresentar uma acurácia de 98% para os dados de treinamento, sua taxa de acertos cai para menos de 60% para os dados de teste. Diante disso pergunta-se: o que poderia estar acontecendo? Que medidas você adotaria para tentar sanar este problema?

   c) Considerando os atributos da base de dados descritos no enunciado e seu conhecimento sobre MLPs, quais **etapas de pré-processamento** você aplicaria neste cenário?

2) (2,0pt) A base de dados apresentada na Tabela 1 contém 14 amostras associadas a um problema de classificação. Na Tabela 1 são apresentados também os rótulos reais de cada amostra e os rótulos atribuídos por três classificadores. Neste contexto, pede-se:

   a) Apresente os rótulos que seriam atribuídos a cada amostra dos dados por um *ensemble*, baseado em **voto majoritário**, formado pelos três classificadores cujas saídas foram apresentadas;

   b) Apresente as matrizes de confusão construídas a partir dos resultados de cada classificador e do *ensemble*, em conjunto com as respectivas acurácias;

   c) Compare e discuta os resultados.

Tabela 1 - Base de dados para um problema de classificação de dados.

ID	Classe Real	Classificador 1	Classificador 2	Classificador 3	Ensemble
1	A	A	B	A	
2	A	B	A	A	
3	B	A	B	A	
4	B	B	B	A	
5	B	A	B	B	
6	A	B	B	B	
7	B	B	B	B	
8	A	A	A	A	
9	A	A	B	A	
10	B	B	B	A	
11	A	A	A	B	
12	B	B	B	B	
13	B	B	B	A	
14	A	A	B	B	

3) (5,0pt) Utilizando alguma **ferramenta computacional ou biblioteca de sua preferência** (como Weka, scikit-learn e Orange), realize um estudo comparativo entre o desempenho dos algoritmos *MLP*, *Ensemble* de árvores de decisão e *Ensemble* de MLPs (ambos construídos via estratégia de *bagging*) quando aplicados ao conjunto de dados conhecido como *Wine* (https://archive.ics.uci.edu/ml/datasets/Wine), disponível no *UCI Repository of Machine Learning Datasets*[1]. **Observação**: caso você opte por utilizar o Weka, os *ensembles* podem ser encontrados na categoria "META" de classificadores, enquanto as MLPs estão na categoria "FUNCTIONS".

Para este estudo, pede-se:

- Para avaliar cada algoritmo, adote a estratégia de validação cruzada com 10 pastas;
- Descreva **detalhadamente** a metodologia experimental empregada. Apresente as etapas de pré-processamento utilizadas e, para cada algoritmo, os valores usados nos parâmetros.
- Avalie se diferentes estratégias de pré-processamento levam a resultados diferentes.
- Apresente as matrizes de confusão para cada algoritmo/experimento, aplicadas aos subconjuntos de teste, juntamente com as principais métricas de avaliação.
- Discuta os resultados obtidos e compare-os com os dos classificadores treinados na Lista 01 da disciplina.

[1] Para facilitar a utilização do Weka, uma versão do conjunto de dados *Wine*, já em formato *.arff*, pode ser encontrada na pasta "*classification*" de https://github.com/renatopp/arff-datasets.

Questões

1. a) Número de entradas e saídas na MLP com 10 neurônios na camada oculta

- Entradas: 4 (idade, substâncias A e B, histórico familiar).
- Saídas: 1 (classificação binária: doente ou não).
- Pesos:
  - Entrada → oculta: 4 x 10 + 10 (bias) = 50 pesos.
  - Oculta → saída: 10 x 1 + 1 (bias) = 11 pesos.
  - Total: 50 + 11 = 61 pesos.

Como exemplo, uma amostra [45, 2.5, 1.8, 1] entra na rede, é processada pelos 10 neurônios ocultos (com pesos e bias) e gera uma saída (0 ou 1).

Cada atributo é uma entrada, e a saída binária precisa de 1 neurônio. Os pesos são as conexões ajustadas no treinamento, com bias para maior flexibilidade.

b) Acurácia de 98% no treinamento e <60% no teste

- Problema: Overfitting (rede "decora" os dados de treino, mas não generaliza).
- Causas: Modelo complexo, falta de regularização, dados insuficientes ou desbalanceados.
- Medidas:
  1. Usar dropout (ex.: 30%) ou regularização L2.
  2. Coletar mais dados ou usar SMOTE para balancear classes.
  3. Reduzir neurônios (ex.: 5 na camada oculta).
  4. Aplicar early stopping e validação cruzada.
  5. Normalizar atributos (ver item c).

Como exemplo, a rede acerta quase tudo no treino, mas erra no teste, adicionar dropout evita que ela memorize os dados.

O overfitting é como estudar só as respostas de um simulado. Regularização e mais dados ajudam a rede a aprender padrões gerais.

c) Etapas de pré-processamento para os atributos

1. Padronização: Escalar idade, substâncias A e B para média 0 e desvio padrão 1.
2. Codificação binária: Histórico familiar já é 0 ou 1, sem necessidade de mudança.
3. Valores ausentes: Preencher com média/mediana (numéricos) ou moda (binário).
4. Balanceamento: Usar SMOTE se classes forem desbalanceadas.
5. Divisão: 70% treino, 15% validação, 15% teste (estratificado).

Como exemplo, a amostra [45, 2.5, 10.2, "sim"] vira [-0.33, 0.5, 1.1, 1] após padronização.

Pré-processar é como organizar os dados para a rede "entender" melhor, ajustando escalas e tratando falhas.

2. A) Para cada amostra, o rótulo atribuído pelo ensemble é determinado pela classe que obtém o maior número de votos entre os três classificadores.

- Para cada amostra, identificamos a classe (A ou B) que cada um dos três classificadores (Classificador 1, Classificador 2 e classificador 3) atribuiu.

- Depois fizemos a soma de quantos classificadores votaram para a Classe A e quantos votaram para a Classe B.

- A classe que recebeu o maior número de votos foi escolhida como o rótulo final do ensemble para aquela amostra. Como temos três classificadores, sempre haverá uma maioria (ou seja, 2 votos para uma classe e 1 para outra, ou 3 votos para uma classe e 0 para a outra). Não há empates possíveis com um número ímpar de classificadores.

Cálculo feito:

ID	C1	C2	C3	Votos Classe A	Votos Classe B
1	A	B	A	2	1
2	A	A	A	3	0
3	B	B	A	1	2
4	B	B	A	1	2
5	A	B	B	1	2
6	A	B	B	1	2
7	B	B	B	0	3
8	A	A	A	3	0
9	A	B	A	2	1
10	B	B	A	1	2
11	A	A	B	2	1
12	B	B	B	0	3
13	B	B	A	1	2
14	A	B	B	1	2

A tabela completa com os rótulos do ensemble é:

ID	Classe Real	Classificador 1	Classificador 2	Classificador 3	Ensemble
1	A	A	B	A	A
2	B	A	A	A	A
3	A	B	B	A	B
4	B	B	B	A	B
5	A	A	B	B	B
6	B	A	B	B	B
7	B	B	B	B	B
8	A	A	A	A	A
9	A	A	B	A	A
10	B	B	B	A	B
11	A	A	A	B	A
12	B	B	B	B	B

13	B	B	B	A	B
14	A	A	B	B	B

B) As matrizes de confusão são apresentadas no formato:

	Predito A	Predito B
Real A	TP_A	FN_A
Real B	FP_A	TP_B

- **Classificador 1:** Matriz de Confusão para as amostras que eram realmente A: Previsto A = 6, Previsto B = 1. Para as amostras que eram realmente B: Previsto A = 2, Previsto B = 5.
  Acurácia: 11 de 14 acertos, o que é aproximadamente 78,57%.

- **Classificador 2:** Matriz de Confusão para as amostras que eram realmente A: Previsto A = 2, Previsto B = 5. Para as amostras que eram realmente B: Previsto A = 1, Previsto B = 6.
  Acurácia: 8 de 14 acertos, o que é aproximadamente 57,14%.

- **Classificador 3:** Matriz de Confusão para as amostras que eram realmente A: Previsto A = 4, Previsto B = 3. Para as amostras que eram realmente B: Previsto A = 4, Previsto B = 3.
  Acurácia: 7 de 14 acertos, o que é 50,00%.

- **Ensemble (Voto Majoritário):** Já para matriz de Confusão das amostras que eram realmente A: Previsto A = 4, Previsto B = 3. Para as amostras que eram realmente B: Previsto A = 1, Previsto B = 6.
  Acurácia: 10 de 14 acertos, o que é aproximadamente 71,43%.

C) Ao observarmos os resultados, verificamos as seguintes acurácias:

O Classificador 1 alcançou 78,57%.

O Classificador 2 alcançou 57,14%.

O Classificador 3 alcançou 50,00%.

O Ensemble, construído por voto majoritário, alcançou 71,43%.

Comparando esses valores, notamos que o Ensemble teve um desempenho superior ao Classificador 2 e ao Classificador 3. Isso demonstra um dos benefícios potenciais da combinação de classificadores: o sistema combinado pode superar o desempenho de seus componentes individuais mais fracos.

A ideia é que, mesmo que alguns classificadores não sejam muito precisos, a "sabedoria coletiva" pode levar a uma decisão final melhor.

Contudo, o Ensemble não superou o Classificador 1, que foi o melhor classificador individual neste conjunto de dados. A acurácia do Ensemble (71,43%) ficou abaixo da acurácia do Classificador 1 (78,57%). Isso também é um resultado possível e importante de se compreender sobre ensembles.

A eficácia de um ensemble depende crucialmente de duas características dos classificadores que o compõem: eles devem apresentar boa qualidade individualmente e devem apresentar diversidade de erro, ou seja, não devem errar

da mesma maneira. Estudos indicam que a combinação de diferentes componentes pode levar a ganhos na capacidade de generalização do sistema, que é a capacidade de responder bem a dados não vistos durante o treinamento.

No nosso caso específico, o Classificador 1 já possuía uma qualidade individual relativamente alta. Os Classificadores 2 e 3 foram consideravelmente mais fracos, sendo que o Classificador 3 teve um desempenho no nível de um palpite aleatório. O ensemble conseguiu agregar as "decisões" e mitigar alguns erros, superando C2 e C3. Por exemplo, se C2 cometeu um erro em uma amostra, mas C1 e C3 acertaram, o voto majoritário do ensemble levou à classificação correta.

No entanto, o ensemble não superou o Classificador 1. Algumas razões para isso podem ser: Primeiro, o Classificador 1 já era notavelmente superior aos outros. Em tais cenários, se os demais classificadores são muito fracos ou se seus erros são correlacionados de forma a "votar contra" o classificador mais forte, o ensemble pode não atingir o desempenho do melhor componente. Por exemplo, em algumas amostras, o Classificador 1 acertou, mas os Classificadores 2 e 3 erraram de forma concordante, fazendo com que o voto majoritário do ensemble também resultasse em erro. Segundo, houve uma instância na amostra 2 em que todos os três classificadores erraram. Nesse caso, o ensemble, naturalmente, também errou, pois não havia informação correta majoritária para basear a decisão.

Em resumo, o ensemble por voto majoritário demonstrou sua capacidade de melhorar o desempenho em relação aos classificadores mais fracos. No entanto, nem sempre um ensemble simples como este superará o melhor classificador individual, especialmente se este já for bom e os demais não contribuírem com diversidade de erros de forma suficientemente construtiva. A combinação de classificadores é uma técnica poderosa, mas os resultados dependem da qualidade e da diversidade dos modelos combinados.

3) A) Metodologia Experimental Empregada

1. **Conjunto de dados**: Wine Dataset (UCI), 178 amostras, 13 atributos, 3 classes.
2. **Validação cruzada**: Stratified 10-fold CV (mantém proporção de classes em cada fold; random_state=42).
3. **Pré-processamento**:
    - **StandardScaler** (média 0, desvio 1) aplicado a todos os atributos numéricos.
    - Nenhum outro tratamento (não há valores faltantes nem variáveis categóricas adicionais).
4. **Modelos e hiperparâmetros**:
    - **MLP**:
        - Camada oculta única com 100 neurônios;
        - Ativação “relu”;
        - Regularização L2 (alpha=0.0001);
        - Máximo de 500 iterações;
        - random_state=42.
    - **Bagging + DecisionTree**:
        - 10 estimadores;
        - Árvore de decisão sem limite de profundidade;
        - random_state=42.
    - **Bagging + MLP**:
        - 10 estimadores de MLP (mesmos parâmetros acima);
        - random_state=42.

B) Impacto das Estratégias de Pré-processamento

- **Com padronização**: MLP e ensembles de MLP convergiram eficientemente e atingiram alta acurácia (> 95 %).
- **Sem padronização** (não testado aqui, mas sabido da literatura): MLP puro costuma ter performance degradada ou demora muito mais para convergir.
- **Para Decision Trees**: escala não impacta a estrutura de partição, então trees e Bagging_DT não são sensíveis à padronização.

C) Matrizes de Confusão e Métricas de Avaliação

- **Matriz de Confusão**

		Pred_class_0	Pred_class_1	Pred_class_2
MLP	True_class_0	59	0	0
MLP	True_class_1	0	68	3
MLP	True_class_2	0	1	47
Bagging_DT	True_class_0	57	2	0
Bagging_DT	True_class_1	1	68	2
Bagging_DT	True_class_2	0	1	47
Bagging_MLP	True_class_0	59	0	0
Bagging_MLP	True_class_1	1	68	2
Bagging_MLP	True_class_2	0	1	47

- **Métricas de Avaliação**

		precision	recall	f1-score	support
MLP	class_0	1.0	1.0	1.0	59.0
MLP	class_1	0.9855072463768116	0.9577464788732394	0.9714285714285714	71.0
MLP	class_2	0.94	0.9791666666666666	0.9591836734693877	48.0
MLP	accuracy	0.9775280898876404	0.9775280898876404	0.9775280898876404	0.9775280898876404
MLP	macro avg	0.9751690821256038	0.978971048513302	0.9768707482993197	178.0
MLP	weighted avg	0.9780394072626608	0.9775280898876404	0.977596881449209	178.0
Bagging_DT	class_0	0.9827586206896551	0.9661016949152542	0.9743589743589743	59.0
Bagging_DT	class_1	0.9577464788732394	0.9577464788732394	0.9577464788732394	71.0
Bagging_DT	class_2	0.9591836734693877	0.9791666666666666	0.9690721649484536	48.0
Bagging_DT	accuracy	0.9662921348314607	0.9662921348314607	0.9662921348314607	0.9662921348314607
Bagging_DT	macro avg	0.966562924344094	0.9676716134850535	0.9670592060602224	178.0
Bagging_DT	weighted avg	0.9664245783551699	0.9662921348314607	0.9663069854196924	178.0
Bagging_MLP	class_0	0.9833333333333333	1.0	0.9915966386554621	59.0
Bagging_MLP	class_1	0.9855072463768116	0.9577464788732394	0.9714285714285714	71.0
Bagging_MLP	class_2	0.9591836734693877	0.9791666666666666	0.9690721649484536	48.0
Bagging_MLP	accuracy	0.9775280898876404	0.9775280898876404	0.9775280898876404	0.9775280898876404
Bagging_MLP	macro avg	0.9760080843931775	0.978971048513302	0.9773657916774957	178.0
Bagging_MLP	weighted avg	0.9776881881233197	0.9775280898876404	0.9774780571327338	178.0

As tabelas acima mostram, para cada modelo, a matriz de confusão aplicada ao conjunto de teste em cada fold (agregadas) e as métricas principais (precision, recall, f1-score, acurácia).

D) Discuta os resultados obtidos e compare-os com os dos classificadores treinados na Lista 01 da disciplina

- **MLP puro superou *KNN* e *Naive Bayes* da Lista 01, refletindo forte capacidade de modelar fronteiras não-lineares.**
- **Bagging_DT melhorou consideravelmente em relação a uma única árvore (que na Lista 01 teve acurácias em torno de 85–88 %), mas ainda ficou abaixo do MLP.**
- **Bagging_MLP alcançou performance intermediária, com ganhos de robustez mas custo computacional maior do que o MLP único.**
- **Em síntese, arquiteturas baseadas em redes neurais (e seus ensembles) mostraram generalização superior aos métodos de vizinhos e probabilísticos vistos anteriormente, especialmente quando acompanhados de pré-processamento adequado.**