Materiais:

Q: Onde encontro os materiais de apoio e os arquivos notebooks para execução dos exercícios?

R: O material de apoio consta no ícone de folha a4 ao lado do título da disciplina.

Q: Há material prático para esta disciplina?

R: Sim, o professor Martin elaborou uma aula extra para aplicar os conceitos vistos em aula de forma prática utilizando alguns algoritmos em Python. O material encontra-se disponível no icone A4 ao lado do título da disciplina, na aula extra.

Ferramentas:

Conceitos e Exercícios:

Q: No caso do alfa da função de custo (15 minutos de vídeo), por exemplo, que ferramenta poderia ser usada para calculá-lo? É possível obter esse dado usando Sklearn, por exemplo?

R: A taxa de aprendizado (alfa) regula a intensidade das atualizações de pesos durante o processo de otimização. Trata-se de um hiper parâmetro, ou seja, não é aprendido durante o treinamento, e deve ser informado pelo usuário antes do processo iniciar. Geralmente utiliza-se valores pequenos, entre 0.1 e 0.00001. O valor apropriado para a taxa de aprendizado depende de diversos fatores, como: qual técnica de aprendizado será utilizada (regressão linear/logística, rede neural etc.), a complexidade do modelo (ex.: profundidade de uma rede neural), função de custo utilizada, entre outros. Para se obter uma estimativa da qualidade deste valor de hiper parâmetro, podemos utilizar algum protocolo e medida de avaliação (aula 6, parte 3 e 4) para testar diversos valores e encontrar valores apropriados. Falando especificamente no Scikit-Learn, recomendo a leitura deste material para entender as funções disponíveis: https://scikit-learn.org/stable/model_selection.html.

Q: A fórmula apresentada na aula 2 parte 3 no instante 37:51 do vídeo contém o elemento "-1" no denominador? A fórmula certa é realmente (rank - 1) / (Qtd.Valores - 1)?

R: Refazer o exemplo do slide para que a explicação fique mais clara.

Categorias = {pequeno, médio, grande}

Fórmula: (rank - 1) / (num_valores - 1)

Assumindo que os ranks são 1, 2 e 3 para pequeno, médio e grande, respectivamente:

- Pequeno: (1 - 1) / (3 - 1) = 0 / 2 = 0
- Médio: (2 - 1) / (3 - 1) = 1 / 2 = 0.5
- Grande: (3 - 1) / (3 - 1) = 2 / 2 = 1

Q: Na aula 2, parte 4, no instante 26:35 do vídeo o professor fala que a coluna " *Weighted distance (1/d)* " é o resultado do valor na coluna ** _Distance L2_ * *(1/*

("Distance L2")) **, desculpe se estou deixando passar algo, mas essa conta não resulta em 1? Aparentemente o que consta na coluna *"Weighted distance (1/d)"* é apenas a conta *(1/(Distance L2))*.**

Para esse método de *voto ponderado* seria apenas *(1/(Distance L2))* e somar os resultados, tendo como vencedor o de maior pontuação?

R: Sim, sua interpretação da coluna "*Weighted Distance*" está correta, ela reflete somente o resultado do inverso da distância.

Agora sobre o cálculo de voto ponderado, você também está correto: bastaria somarmos os resultados dos inversos das distâncias para cada classe e verificar qual das classes possui a maior pontuação. Contudo, este resultado não representa uma probabilidade, pois não estaria normalizado. Para obter uma interpretação probabilística como saída, bastaria dividir a soma de cada uma das classes pela soma total de todos os inversos das distâncias.

Pegando como exemplo a tabela do slide 62:

- Benign: 2 (0.5)
- Malignant: 2.2360 (0.448)
- Malignant: 2.8284 (0.354)

Soma total dos inversos das distâncias: 0.5 + 0.448 + 0.354 = 1.302

Valor para a classe Benign: 0.5 / 1.302 = 0.3840 (38.40%)

Valor para a classe Malignant: (0.448 + 0.354) / 1.302 = 0.6159 (61.59%)

Q: Tempo 28:03. Achei que seria escolhido a menor distância (mais similar) e não a maior distância (menos similar), o certo seria definir como Benigno como está no slide não é mesmo?

Só seria escolhido a classe com a menor distância para o caso k=1. Para os demais casos (k>1), devemos contar o número de votos para cada classe, e escolher a classe com o maior número de votos. No caso do voto ponderado, bastaria somarmos os resultados dos inversos das distâncias para cada classe e verificar qual das classes possui a maior pontuação. Neste caso, o maior valor é o da classe Malignant (0.802). Contudo, este resultado não representa uma probabilidade, pois não estaria normalizado. Para obter uma interpretação probabilística como saída, bastaria dividir a soma de cada uma das classes pela soma total de todos os inversos das distâncias.

Pegando como exemplo a tabela do slide 62:

- Benign: 2 (0.5)
- Malignant: 2.2360 (0.448)
- Malignant: 2.8284 (0.354)

Soma total dos inversos das distâncias: 0.5 + 0.448 + 0.354 = 1.302

Valor para a classe Benign: 0.5 / 1.302 = 0.3840 (38.40%)

Valor para a classe Malignant: (0.448 + 0.354) / 1.302 = 0.6159 (61.59%)

Q: Professor, eu sempre posso utilizar a matriz de confusão como acurácia? para qualquer caso?imagino que nos casos com poucos dados, mas é sempre confiável usá-la?

R: A matriz de confusão em si não é uma métrica, mas sim um mecanismo usado para extrair informações sobre as saídas do modelo e extrair métricas de desempenho a partir disto. Como discutimos em aula, as métricas que devem ser usadas dependem de uma análise do contexto do problema. Por exemplo, se as classes do problema estão balanceadas, seria viável utilizar a acurácia. Mas é possível extrair outras medidas à partir da matriz de confusão, como precisão, revocação, etc.

Q: Como calculo via código as outras medidas de avaliação como: precision, recall e F-mesure?

R: Nesse caso, você pode utilizar uma biblioteca do scikit-learn mesmo, que você passa o label predito e o label real e a lib mesmo lhe calculará. Segue o link da mesma caso queira tentar: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

Complementando: o y_true você já possui, pois é o array contendo os rótulos do seu conjunto de TESTE\VALIDAÇÃO. O y_pred se refere ao array com as PREDIÇÕES do seu modelo treinado anteriormente (classifier.fit). Nesse caso, você precisa fazer a predição agora com seu conjunto de dados de validação\teste chamando o método predict do seu modelo. Esse método retornará um array de resultados que será o seu y_pred.

Q: Nos slides da aula 02, há referencia a bibliografia:[Mitchell 1997] Capítulo 1, [Tan et al. 2006] Capítulo 1, mas sem os nomes específicos dos livros em questão estou tendo dificuldade de encontrar o material.

R: Quanto a referência de Mitchell 1997 é: Mitchell, Tom M. 1997. Machine Learning. First. McGraw-Hill Science/Engineering/Math. (https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf) A referência de Tan et al. 2006 é sobre o livro de Tan "Introduction to Data Mining" (https://www-users.cse.umn.edu/~kumar001/dmbook/sol.pdf)

Q: A rede neural perceptron funciona apenas com atributos preditivos binários? Poderiam ser números reais?

R: O multi-layer perceptron pode ser utilizado para classificação (categorias, não necessariamente apenas binário) e também para regressão (predição de valores) MLPRegressor. É possível verificar como utilizar na documentação do Scikit Learn também.

Q: Por que "a gente quer" Theta transposto multiplicado por x?

R: Este assunto é discutido em todas as aulas de aprendizado baseado em otimização (aulas 4 a 6). Talvez o momento de explicação mais relevante seja na aula 4 (slides 63-73). Respondendo de maneira genérica, "Theta" representa os pesos de um modelo e "x" representa a entrada do modelo. É necessário transpor Theta para que a multiplicação entre matrizes/vetores seja válida.

Por que quando ela ultrapassa a margem não importa?

R: Considerando que a referência é Support Vector Machine (SVM), O conceito de margens (hard e soft) e vetores de suporte são explicados no início da aula 5. Estes conceitos são importantes para entender o porquê de instâncias que ultrapassam a margem (i.e., ficarem mais distantes da fronteira de decisão) "não importarem" durante o treinamento. O próximo conceito relevante, também explicado na aula 5, é a função de custo de SVMs, chamada de Hinge Loss. Esta função de custo utiliza um limiar para decidir o custo de uma

instância de acordo com sua distância para com a margem. As instâncias que geram custo adicional no modelo são aquelas que estão dentro das margens (support vectors), e usamos um hiperparâmetro C para definir o quão intensamente o modelo deve ser penalizado por causa destas instâncias. Quanto maior C, mais penalizado o modelo será por erros de classificação (hard vs soft margin).

Qual a aplicação prática de uma fronteira de decisão?

R: O conceito de fronteira de decisão é abordado em praticamente todas as aulas, com exemplos práticos. No caso de redes neurais, por exemplo, uma fronteira de decisão é gerada através dos parâmetros do modelo e separa o espaço d-dimensional dos atributos entre classes pré-definidas. Uma fronteira de decisão é usada para decidir a qual classe cada instância pertence, de acordo com o modelo.

Q: Porque, na regularização do hiper parâmetro, ao aplicar o lambda não queremos regular o bias? Qual o exemplo prático que demonstra a razão pela qual eu não quereria ter isto? Na prática, como isto afeta a análise das classes?

R: O início da aula 4 contém uma explicação relevante para esta pergunta. No início desta aula, o professor Martin explica o impacto do bias e dos pesos usando a equação da reta como analogia. Não podemos regularizar os parâmetros livres (biases) pois eles dão liberdade para o modelo ajustar suas predições de maneira independente dos dados de entrada. Em regressão linear, por exemplo, isto nos permite "deslocar" a reta de predição para que ela não precise obrigatoriamente cruzar a origem do plano cartesiano (o que poderia ser ruim em determinados conjuntos de dados), facilitando o processo de aprendizagem do modelo. De maneira similar, em regressão logística os termos livres nos proporcionam a possibilidade de criar fronteiras de decisão que não cruzem a origem.

** FAQ gerado com base em comentários até o dia 30/11/2022.*