BLOCO 1

Boa noite, pessoal! Tudo bem com vocês? Terçou e terçou com a aula de DSA do professor Juca, né? Aula de Árvores, Redes Neurais e Icembol, é isso aí. Pessoal, antes de a gente iniciar a aula e eu passar a palavra para o professor Juca, eu quero agradecer a nossa equipe. E nós temos aí a Mari, que está cuidando da nossa transmissão. Mari, muito obrigada! Temos também o Gui, que está aí dando um apoio para nós também. Então, qualquer problema que vocês venham a ter com transmissão, pode mandar lá na aba de perguntas, que o Gui vai dar todo o auxílio para vocês, tá bem? E também temos os nossos professores, né? Que estão à disposição de vocês. Então, conforme o professor Juca for passando o conteúdo, surgiu alguma dúvida na aba de perguntas, tá, pessoal? Então, teremos lá com a gente o professor Renato, o professor Fabiano, o Fernandinho Freire, o professor Marcelo Sabadini, o Alexandre Duarte, o Igor e o Tiago Almeida, tá bem? E também o time aí de mediação, Lorisa, Maria Karina e a Cláudia Cacau, que está aqui com a gente hoje também. Olha, pessoal, então eu desejo para vocês uma excelente aula. Quero saber. Quem está assistindo as defesas de DSA, conta aí para a gente. Alguém assistiu? O que vocês acharam? Gostaram de um tema específico? Compartilha com a gente, tá bem? Professor Juca, boa noite! Seja muito bem-vindo. Como passou? Olá, tudo bem? Ah, não, achei que eu estava no mudo. Foi ao contrário. Em geral, a gente fala e esquece de desligar o mudo. E aí, pessoal, tudo joia? Foi bem de semana. E vamos começar aqui a nossa segunda aula de árvores. Hoje tem bastante coisa para a gente falar. A gente vai falar de como é que a gente usa o algoritmo para quando tem uma resposta contínua. A gente vai entrar em outros algoritmos. E uma coisa mais básica. Quando eu falo básica, não quer dizer que é fácil. Quer dizer que é base. E base é uma coisa que você usa para construir outros conhecimentos. E que é de validação cruzada. É um aspecto muito básico de aprendizado de máquina. É básico porque, de novo, é importante para um monte de outras coisas. E é isso que a gente vai ver hoje. Estou super animado aqui. Bom, acho que você ia falar mais alguma coisa. Ju, então está tudo comigo aqui. Vamos lá, vamos começar. Estou animada. Deixa eu compartilhar minha tela aqui. Bom, vamos lá. Ligar o meu apontadorzinho aqui. Então, a gente está na segunda aula da nossa aula de árvores, redes e ensemble models. Bom, um disclaimer aqui. Padrão, né? Já vamos se preparando. Abre o Spyder. Arruma alguma coisa para anotar. Quem gosta de anotar em físico, quem gosta de anotar em eletrônico, se organiza. Pode abrir o projeto. Sempre, em geral, eu falo para começar abrindo um script 00. Mas hoje não tem nada de muito diferente no nosso projeto. Então, vocês vão ver que... Ele não tem um script 00 aqui. Fechou o meu projeto aqui. Bom, eu vou abrir de novo. Só um segundo que isso... Eu não sei por que ele fechou o meu projeto aqui. Agora sim. Então, não tem o script 00. E aí a gente pode já deixar aberto o primeiro script aqui. Que é a solução da lição de casa, né? Que a gente deixou lá para... Vocês lembram? Era um arquivinho, uma base de dados que eu deixei um script que simula ela. Aí dá até para alterar. Dá para ser sempre legal ter a mão em uma forma de simular uma base de dados. Mas uma base de um modelo de scoring de crédito. Que é uma coisa bastante popular desde os anos... 70, eu diria. Já pode deixar aí tudo preparado. E a gente já volta para o Spyder. Aliás, quem conseguiu mexer no exercício, conta aí para a gente. Teve dificuldade? Conseguiu? Deu tempo? Não deu? Porque eu acho que é legal... Assim, né? A gente potencializa o aprendizado quando a gente tem contato com o conteúdo mais de uma vez. No período, assim, né? Então, por isso que eu deixo o exercício. Bom, então vamos dar uma recapitulada. A gente falou sobre o que era Machine Learning. Inteligência Artificial. Falamos do paradigma Machine Learning e de nomenclaturas. Então, Machine Learning. Nossa definição eram agentes que executavam... Que pegavam percepções ou inputs e davam outputs. Ou previsões, né? Bom, a Inteligência Artificial é um ramo que visa executar tarefas normalmente feitas por seres humanos. Dos mais diferentes tipos hoje em dia, né? E o paradigma Machine Learning em contraposição... Assim, né? Mais um mindset, né? Em contraposição a uma abordagem estatística. A abordagem estatística. A gente... Supõe que a distribuição é normal. Ou T, ou sei lá o que for. Que os dados têm uma certa estrutura de correlação. A gente supõe um monte de coisas e faz o nosso trabalho. No Machine Learning, a gente não supõe nada. A gente só está buscando padrões nos dados. Por um lado, isso nos deu algoritmos poderosíssimos aí de aprendizado de máquina. E por outro... A gente não consegue responder aquela pergunta, né? Que o pesquisador, o médico faz. Assim, ah, será que... Praticar atividade física é fator de proteção contra determinada doença? Aí ele pode até ver um padrão nos dados. Mas ele não consegue fazer, assim, né? Filosoficamente mesmo. Ele não consegue falar que a relação existe. Controlando o erro. É isso que o estatístico faz. Ele controla o erro da afirmação dele. Ele fala, né? A média deve estar entre tanto e tanto com confiança 95%. Isso quer dizer que ele está controlando o erro tipo 1 da afirmação que ele faz. No Machine Learning, a gente não está preocupado com nada disso. A gente só está preocupado em achar um padrão. Ver se ele é útil. E botar ele para funcionar. Bom, a gente falou um pouquinho das nomenclaturas, né? Vamos dar uma olhadinha na árvore que a gente viu na aula passada, né? Cada pedacinho, né? Cada quadradinho aqui é um nó da árvore. A gente pode chamar de nó. Os nós finais. A gente chama de folhas. E uma quebra. É uma regra, né? É uma quebra. Ou é um... É uma nomenclatura em inglês, né? É um split. Aí, a nomenclatura em inglês é bom saber que vira e mexe aparece lá no nosso código, no help. Então, a árvore a gente tem 38% de sobreviventes no Titanic e 100% da base. A primeira perguntinha que a gente faz. O cidadão é do sexo masculino? Sim. Vem para cá. Então, aí ele tem 19%. Cai a probabilidade de... Sobrevivência dele. A idade é maior ou igual a 6,5 anos? Ou seja, ele é criança? Se a resposta... Ele é adulto, aliás, né? Se a resposta é sim, vem para cá e cai ainda mais a probabilidade de sobrevivência. Com uma boa parcela ainda da amostra aqui, né? 62%. É da segunda ou terceira classe? Sim. Então, a gente vem para essa última... Última folhinha aqui, né? Bom, então, depende se ele não for da segunda ou terceira, ou seja, da primeira classe, a probabilidade de sobrevivência é um pouquinho maior. Ou, por outro lado, um passo antes. Se é homem e é criança, a probabilidade de sobrevivência é bem maior, mas tinha pouquinho menino, né? Homem e criança no navio, só 3%. Da população. E daí, por outro lado, vão estar as mulheres. E assim, a gente vai cobrir qualquer combinação dessas features aqui, né? Dessas variáveis que a gente usou para construir essa árvore. E a gente vai ser capaz de dar uma probabilidade de sobrevivência para cada passageiro do navio ali. Dá para fazer árvores melhores que essa, mas essa acho que é uma árvore bem... Inicial, assim, né? Bem óbvia, talvez, né? Para a gente começar. Então, a gente falou do que era overfitting. O overfitting, a gente tem uma ideia do underfitting. A gente tem um padrão mais... A gente tem a possibilidade de captar um padrão mais complexo de melhorar o acerto, né? Tem um padrão normal, equilibrado, que a gente busca. E um overfitting que a gente classifica tão bem a nossa base de treino que a gente pega particularidades dela que não vão ser replicáveis em outras populações, né? Eu lembro uma piada do Big Bang Theory, que o cara fala Ah, pensei numa solução para determinado problema. Ah, mas só funciona com galinhas perfeitamente esféricas no vácuo. Então, não vai dar certo. Bom. Adorava aquela série. E aí, a gente viu o cross-validation, né? A primeira ideia, a ideia mais básica do cross-validation, a gente tem a nossa base, a nossa amostra total disponível. A gente quebra ela na amostra de treino e a amostra de teste. A gente desenvolve o modelo aqui, treina, né? O que é treinar o modelo? É escolher as quebras que a árvore vai fazer, né? A gente escolhe as quebras, as quebras otimizando na base de treino. Depois, a gente vai avaliar o nosso modelo na base de teste mesmo, tá? Essa ideia mais básica, a gente vai ver uma ideia um passo além daqui a pouquinho. Que, aliás, foi uma outra coisa que eu deixei para vocês derem uma olhada, né? O que era dar uma bugada e o que seria o algoritmo do CAFOLD, que é o que a gente vai ver aqui na sequência. Então, a gente viu que a gente não precisa fazer o corte de 50% na árvore, né? O corte de 50% seria igual a essa linha aqui. Essa tabelinha aqui, para lembrar, ela dá um menos a especificidade e a sensibilidade do nosso algoritmo para cada possível ponto de corte. O que é o ponto de corte? Ah, o básico, o default é se a probabilidade de sobreviver, sobreviver é maior que 50%. Eu vou dizer que o cidadão é, vou classificar ele como sobrevivente, menor que 50% vou classificar como não sobrevivente. Por que o 50%? Ah, porque intuitivamente ele expressa o não sei, né? Assim, ele traz uma ideia de imparcialidade. Então, a gente pode usar outros pontos de corte. Para cada possível ponto de corte, a gente vai ter, ou menos, a especificidade. A especificidade e a sensibilidade, né? A sensibilidade é a probabilidade da gente pegar um evento, um sobrevivente e classificar ele como sobrevivente. A especificidade, pegar um não evento, classificar ele como não evento, um não sobrevivente e classificar como não sobrevivente mesmo. Assim, né? Importante, classificar ele corretamente, dado que ele é não sobrevivente. É condicional. Essa é a característica. Daí, no primeiro ponto de corte, eu classifico todo mundo como evento, sobrevivente. Um menos a especificidade é um erro. Então, os não sobreviventes, eu errei todos. Os sobreviventes, eu acertei todos. Então, eu estou nesse ponto aqui. Aliás, eu estou nesse ponto aqui. E aí, por outro lado, se eu classificar... Se eu classificar todo mundo como não evento, eu vou ter o oposto. Eu vou errar todos os eventos e eu vou acertar todos os não eventos. Eu vou estar nesse ponto aqui. A sensibilidade é zero e um menos a especificidade é zero também. Conforme a gente vai alterando o ponto de corte, eu vou subindo a sensitividade, que é o acerto, e eu vou aumentando um menos a especificidade, só que supostamente em uma velocidade menor. Então, quanto a gente vai alterando o ponto de corte, quanto mais gordinha a curva, melhor o modelo. Daí surge a área de baixo da curva ROC, que é o AUC, uma medida importante para a gente usar no Machine Learning, num problema como esse de classificação. Então, vamos lá. Vamos dar uma olhadinha na nossa atividade aqui. Eu deixei o exercício 1, uma solução. Uma solução. Tem várias soluções. Ju, está na escuta? Câmbio. Pronto. Estou sim. Pode perguntar e eu também preciso de um favor seu. Sim. Pode falar. Pode ficar aparecendo a barrinha do Zoom na sua tela. Você consegue minimizar ela para nós? Isso aqui? Isso. É uma setinha que tem. Eu tiro para cá. Pronto. Melhorou? Não, ainda está aí. É uma flechinha que tem no canto verde. Está aparecendo a gente aí. Nossa, não estou vendo, não. Deixa eu ver no retorno aqui. Deixa eu ver se agora foi. Agora está tudo ok, Prô. Está bom. Eu ia te falar se o pessoal respondeu, quem fez o exercício. Responderam. Posso ler para você, Prô? Por favor. Então, vamos lá. Vamos começar lá por Vitória, no Espírito Santo. O Benjamin disse que ao estudar a aula da semana passada, ele ficou com algumas dúvidas sobre o conceito de custo, de complexidade. Ele chegou a rodar a árvore para as diferentes profundidades, porém o CCP sempre foi o mesmo. Não sabe se ele errou ou se isso faz sentido. Calma aí. Repete o que ele fez. Ele rodou... Ele rodou a árvore para diferentes profundidades. Ah, sim. Tá. Pode fazer sentido, porque o CCP Max é uma medida de complexidade. E aí é diferente da profundidade. A profundidade é uma medida de complexidade também. Mas se... Se a gente olhar, por exemplo, aqui na arvorezinha do Titanic, essa folha aqui, ela começa a quebrar pouquinha gente, né? E aí ela pesa mais do que essa folha, por exemplo, que tem 49% da base. Então, a complexidade leva em conta a profundidade, o número de observações que tem, ela leva em conta... É um parâmetro um pouquinho... É um parâmetro um pouquinho mais completo do que a profundidade. Então, acho que pode sim, dependendo do range de profundidades, pode dar diferente. Mas a ideia é pegar todos os possíveis CCP, porque cada folha está associada a um custo de complexidade limite para ela ser aceita. Aí o algoritmo dá cada nó, na verdade. O algoritmo dá todas essas complexidades. Todos esses custos, né? Qual é o custo que eu vou ter uma árvore com uma folha só? Se tem um custo bem alto ali, ele vai podar a árvore inteira. Dependendo se eu deixar um custo de complexidade zero, ele vai deixar a árvore ser feliz e ela vai se completar ali, né? Vai fazer tudinho. Então, a ideia é deixar a árvore ser feliz, pegar todos os possíveis CCP max e depois a gente testa eles. A gente já vai mexer. A gente já vai mexer aqui no nosso código, né? Vamos lá. Tem mais alguma? Mais algum comentário? Tem, sim. Pronto. Tem mais? Ó, a Valéria disse que adorou que teve lição de casa, que reforçou muito o aprendizado. É isso aí. Aula dada, aula estudada, né, Valéria? É, legal. A Loiane, lá de Brasília, teve um pouquinho de dificuldade, mas ela não mencionou qual foi. Lo, mande para nós qual foi sua dificuldade. A Michelle, professora de São Paulo, tem uma dúvida. O Spyder é melhor que o Jupyter Notebook ou o Google Collab? Ai, olha, a interface, né? Tem várias interfaces super legais e poderosas para a gente usar Python, né? O pessoal mais roots gosta de um que chama Emacs. É um negócio bem roots mesmo, assim, né? Mas ele é bem bom. Ele é bem poderoso no sentido de tecla de atalho. Então, a galera que se acostuma, né, ele é difícil, ganha muita produtividade com ele. Assim, o Jupyter, eu diria que é uma das plataformas mais populares que tem, né? O Google Collab, ele imita o Jupyter Notebook, mas ele é do Google, né? Ele tem umas coisas legais. Tem uma IA lá, que é bonitinha. Assim, nativo, né? Quando eu, uma vez eu dei uma aula com ele, aí ele leu o enunciado, eu deixei uma pergunta num markdown, ele leu o enunciado e já sugeriu a resposta, assim. Não dava nem graça para os alunos pensarem no... Aí eu tinha que mandar desligar a IA para fazer esse tipo de exercício. Ainda que eu ache que IA é um aliado valiosíssimo. Para a gente aprender, programar, para fazer tudo. Mas o Spyder é uma plataforma que ele tem algumas coisas muito legais. Ele imita o RStudio, né? Ele tem umas coisas legais que não vai ter no Jupyter, né? Uma interface bonitinha para você ver o que tem dentro do objeto. Tem umas facilidades aqui, né? Essa disposição da tela é bem diferente no Jupyter. Mas, enfim. Super de preferência. Pode mandar mais, prof? Pode, pode. Manda mais uma. A gente segue. Bom, o Cláudio, o Cláudio lá de Barulhos, perguntou, ele fez o exercício, né? Aí ele colocou um código aqui, professor. Vamos ver se eu consigo ler isso. O AUC 0.92 com XGBoosting, né? Uhum. Barra, AUC 0.80 com o Random Forest, barra, o AUC 0.77 com o modelo Decision Tree, ambos usando o Random Search com combinações diferentes de parâmetros para cada modelo. Ele fez esse exercício, professor. Legal, tá bem completinho, tá bem bacana. Aí dá para a gente ver que o... O XGBoosting que ele rodou ficou melhor que os demais. Parabéns. Vamos lá dar uma olhadinha aqui no nosso exercício. Depois a gente volta com um pouquinho mais. Mais para eu ter a temperatura e se vocês fizeram e tudo, né? Então, vamos lá. Primeiro, a célula pode rodar a lei inteira. A gente está carregando as bibliotecas. Deixa eu ver se tem alguma novidade aqui. Ah, nossa funções ajuda, né? A gente vai usar a descritiva e avaliar a CLF. Descritiva. A mesmíssima função que a gente usou na... Na aula passada. Como ela retorna uma média, funciona tanto para a variável binária quanto para a variável contínua, né? Numérica, quantitativa. Desde que a variável binária seja 0 e 1, né? Aí eu avalio a CLF, um código para ajudar a gente a avaliar a nossa árvore. Scikit-learn, tem test split para a gente fazer o nosso cross-validation. Seja um tree classifier, o algoritmo que a gente vai usar aqui. E a UC score. Eu fiz também um random forest. Ele fica bastante melhor, mas a gente vai dar uma olhada aqui na... Na aula de hoje. É o nosso tema de hoje, o random forest, tá? O Matplotlib, Seaborn. Leu a base aqui do exercício. E uma olhadela na cara. Do banco de dados, né? Bom, o info, a gente tem a variável idade, renda, dívida, utilização de crédito, consultas recentes, inadimplência, que é a nossa target, e a idade categorizada. Então, vamos lá. A gente vai rodar aqui as descritivas, para fazer um roteirinho tão completinho, quanto cabe aqui na aula, né? Então, a primeira variável, idade, a gente quebra ela em cinco grupinhos. Ele quebra na função descritiva. Se a variável é contínua, ela já verifica e quebra em cinco grupos por quantias, né? Aí tem mais ou menos a mesma frequência. E a gente vê que a taxa de inadimplência vai diminuindo conforme aumenta a idade nesta população, neste produto. Nesta base de dados. Ah, e pro quer dizer que quanto mais velho, menos risco de crédito? Não. Eu estou falando que nesse exemplo, aconteceu isso. Até porque é um exemplo simulado, né? Então, aconteceu porque eu determinei que aconteceria. Então, mas para alguns casos, idade tem a ver mesmo com risco de crédito, de alguma forma. Vamos lá. Então, aí a gente vê... Vai separar a variável y, que é a inadimplência. É uma variável tipo 0 e 1, né? 1 quando o indivíduo é inadimplente, 0 quando ele é adimplente. E aí a gente vai separar o x, que é tudo menos a target. Um exemplinho assim bem curtinho, bem direitinho para a gente praticar mesmo, né? E aí uma verificação aqui, né? Tudo que a gente faz, a gente verifica. A mensagenzinha aqui, né? O shape da base x de treino é 8 mil linhas e 6 colunas. Do y treino, 8 mil linhas. Do teste, 2 mil linhas e 6 colunas. E o y é só as 2 mil linhas de uma única coluna. Então, está tudo certinho, tudo coerente, né? Como deveria. Assim. Eu sempre gosto de fazer isso, porque eu já vi gente permutar os valores aqui. Ele vem nessa ordem, né? Esse cara aqui vai ser dividido em dois. Esse e esse. O segundo sendo de acordo com a nomenclatura do teste. Por exemplo, do teste size aqui, né? E esse aqui vai ser dividido entre esse e esse. Nessa ordem aqui. x treino, x teste. N, y teste. Então, se permutar, vai dar ruim. E aí, um random state aqui. 2360873. Meu random seed favorito. Então, a gente roda a nossa árvore aqui. Decision tree classifier. Rodamos. Até agora não fez nada, né? A gente, de novo, importou a classe. CLF agora é um objeto do Python com todas as propriedades. O que são propriedades no Python? Funções e métodos da classe do decision tree classifier. Aliás, métodos e atributos, né? O método é uma função, né? Então, o clf.fit, a gente vai treinar o modelo aqui com o x treino e o y treino. A partir desse momento, o clf já tem as propriedades de uma árvore treinada. A gente pode usar ele para fazer um predict. Pode usar ele para... Verificar as importâncias das variáveis. Um monte de atributos que vai estar lá nele que ele construiu durante o treinamento. Ok. Então, vamos ver como é que ficou nessa... Nossa árvore, né? Então, a gente roda o avalia.clf na base de treino. Vamos lá. A gente gerou essa matrizinha de confusão. Olha que beleza. Lembra o que é isso? O que o modelo falou? O modelo falou que o cara é bom. E o cara é bom mesmo. Quando o cara é mal, ninguém é o modelo. O erro aqui está zero. Quando o modelo falou que o cara é mal, o cara é mal mesmo. Então, o modelo acertou todo mundo na base de treino. Como diria um amigo meu, melhor que isso só se for verdade. Aí, a hora da verdade é na base de teste. Ah, e olha só. Estou quase pulando aqui. O AUC desse modelo ficou um quadradinho perfeito aqui, né? Ele sobe certinho até o 100%. Vai até o outro 100% e cai. Bom, mas na base de teste, a gente tem uns punhadinhos de erro aqui, né? Assim, tem um acerto ok. Mas olha só, que curioso. Quando o cara é mal, eu falo bem mais palpites para o lado do bom. Então, está muito legal isso aqui, né? Não está muito bacana. Olha o AUC dele aqui. Está meio... Então, a gente vai ter que ver se aconteceu alguma coisa com essa árvore. O que eu pus de parâmetro? Nada. Então, a árvore foi feliz da vida, crescendo de acordo com o algoritmo default dela ali. Então, aqui que a gente vai pegar o CCP pass, né? O custo de complexidade... O caminho dos custos de complexidade. De novo, né? Deixa eu resgatar aqui. A árvore é um caminho. Então, a gente tem o caminho dos custos de complexidade. Para um determinado custo, aí vamos dizer, eu tiro essa folhinha aqui. Para um... Eu aumento um pouquinho o custo, talvez eu tire essa folhinha aqui. Aí eu aumento um pouquinho, aí a gente vai cortando a árvore, não necessariamente por camada, né? Então, a gente vai... Às vezes, a gente pode fazer um corte numa profundidade menor por causa do custo de complexidade, né? Então, a gente vai pegar os custos de complexidade do caminho da árvore, os custos limites de cada folha, né? Como que a gente vai fazer? Eu vou transformar num data frame. Onde estão os custos de complexidade? No objeto CLF. Lembra que ele está 3D. Tem os atributos e métodos. Custo de complexidade. O nome da função, do método é Cost Complexity Planning Path. Aí eu indico o X e o Y. E aí ele traz aqui os custos de complexidade da minha base. Ele vai rodar o método e trazer os custos de complexidade. Bom, daí pode rodar esse carinha aqui. E aí, eu vou fazer um... E aí, essa função aqui é para transformar num data frame, né? Senão, a gente não vai conseguir... Vai ficar mais fácil de visualizar, né? E tudo. Vamos dar uma olhadinha aqui onde é que ele está. CCP Path. Dá uma olhada nele. Olha só. Ele tem essa cara aqui. Tem os CCP alfas possíveis. E a impureza. As impurezas possíveis. Você vê que tem alguns números que repetem aqui. Então, um cuidadinho. Que, eventualmente, a gente vai precisar tomar, né? Então, como é que a gente vai fazer aqui? Aqui até a gente está correndo por todo o nosso CCP alfa. Como essa base é pequenininha, ele roda bem. Mas o certo era a gente tirar a duplicidade dos CCP alfas. Vai ter uma certa redundância aqui, né? Então, eu vou iniciar. Gines com uma lista vazia. Para cada custo de complexidade dentro da minha lista, eu vou fazer isso. Eu vou fazer uma árvore com este custo de complexidade, com uma profundidade grande, usando o Gine, um Random State aqui. Eu vou fitar a árvore. Eu vou calcular o ALC da minha árvore. Eu vou calcular o Gine. O Gine da árvore. O ALC, lembra? A funçãozinha, ROCALCscore. Ele calcula já o ALC. A gente põe o Y-Test e os preditos. E ele já calcula o ALC. E aí a gente faz um append do Gine na nossa lista aqui de Gines. Então, a gente rodando tudo isso aqui. Ele realmente demora um pouquinho mesmo. Deixa eu ver. Assim, se alguém quiser deixar mais rápido, acho que dá para fazer um unique aqui. Deixa eu ver. É, o unique. Acho que ele já resolve aqui. Ele faz um pouquinho mais rápido. A gente pegar os valores únicos da nossa árvore. Bom, ele tinha rodado, né? Mas a gente obteve um Gine máximo. É, não. O unique não vai dar certo. Tem que mexer mais aí no negócio, no algoritmo. O unique não vai dar certo. Deixa assim mesmo. Então, o Gine máximo dessa árvore é 54,78. Dessas árvores todas que a gente fez. E o CCP é esse valorzinho aqui. Esse é o custo de complexidade aqui. Então, é tudo que a gente está fazendo aqui. A gente fez um plot. Vamos dar uma olhadinha no nosso plot aqui também. Esse plot está falando, olha como é que é o CCP alfa e o Gine da árvore em função do CCP alfa. O Gine da árvore, deixa eu ver onde é que a gente está calculando aqui. Na base de teste, né? Isso é importante. Então, o Gine da árvore na base de treino, ela é maior quanto maior a complexidade da árvore. Na base de teste, o Gine começa ruim, aí ele escala muito rapidamente até um valor pico. E depois, se eu aumentar mais o custo de complexidade, volta a cair o Gine da árvore, né? Então, não deve ter pontos ao longo disso aqui tudo, né? Deve ter um pontinho aqui e um pontinho aqui só. E aí, dá a impressão, não necessariamente é interpolável essa curvinha aqui. Mas ela dá essa ideia, né? Conforme eu aumento a complexidade da árvore, eu vou ganhando o poder do modelo. Aí, se eu aumentar mais, aí ele começa a pegar características muito particulares da base que acabam atrapalhando na generalização do nosso modelo. Então, a gente vai pegar a árvore ótima. Como, seguindo a cartilha do Tibichirani, que é o meu guru de livro, né? De literário, ele fala para a gente tunar o nosso algoritmo e rodar ele de novo depois no final, né? Até o Tibichirani fala para a gente rodar com a base inteira. Eu estou rodando aqui só na base. E aí, a gente roda a nossa árvore tunada. Eu fitei ela já. Rodei o Decision Tree Classifier e o Fit com o CCP Max que eu peguei lá do passo anterior. Bom, onde que a gente pegou o CCP Max? Acho que foi uma linha que eu passei meio batido, mas uma linha bem tranquilinha. A gente pega... Bom, os genes. A gente criou um data frame de avaliações com CCP, CCP Alpha, Gene. Uma série de quantidades que a gente calculou no FOR. Pegou o Gene Máximo. E o CCP Máximo, ele vai ser o CCP tal que o Gene é o Gene Máximo. E aí... Acho que tem uma função mais bonitinha que essa. Mas essa função é bem funcional. Por assim dizer. Então, vamos lá. A gente rodou a nossa árvore ótima. E aí, a gente avalia ela agora na base de treino e na base de teste. Na base de treino, erra? Erra. Mas acerta bem. O Gene tem uma carinha bem mais saudável. E na base de teste também. Tem um erro bem mais controladinho. E o Gene tá bem parecidinho, né? Ao ser que ele indica que é 77 no teste, 77 no treino também. Igual. Igualzinho. Assim, não precisa ser tão igual, né? O que interessa mesmo é a base de teste. O que manda é a base de teste. Legal? Legal. Então, vamos voltar pra nossa vaca. Fria. Uma expressão que eu nem sei direito onde que vem. Então, bom, exercício a gente já fez um da aula passada. Vamos falar ainda sobre árvores de regressão. Que é o que a gente vai falar agora. Técnicas de validação cruzada. O bagging, né? O mais famoso é o random forest. O boosting tá na aula que vem. E o grid search a gente vai falar. Vamos falar um pouquinho hoje também. Então, sobre árvores de regressão. Eu adoro essa frase aqui do Vicente Matheus, né? Assim, nesse momento aqui é bastante comum a gente se deparar com algumas dificuldades, né? Então, citando o Vicente Matheus, o difícil, vocês sabem, não é fácil. Ele tem várias outras frases geniais também. Era um diligente do Corinthians aí. E... É... Mas, assim, como dizia o meu orientador, tudo que você sabe é fácil, né? E, pegando uma frase do Musashi, um pouco mais séria que essa, tudo no começo é difícil, né? Até respirar no começo é difícil, não é verdade? Então, tudo no começo é difícil. Então, vamos aproveitar a nossa jornada aí da melhor forma possível. Bom, árvore de regressão. Ela é muito semelhante à árvore de classificação. O que muda é o critério de impureza, que é uma coisa baseada no erro. O que é o erro da árvore? O valor verdadeiro menos o predito. Aí a gente eleva ao quadrado e soma tudo. Essa é a soma do quadrado do erro do nosso modelo, tá? Bom, como é que... Cara, que vai ter uma árvore de regressão? Aqui a gente vai fazer esse exercício, que é um exercício, assim, bem para a gente... Pegar a manha do algoritmo, assim, entender, construir uma intuição sobre o que ele está fazendo. Nosso banco de dados tem uma única variável x e uma variável y. Se a gente mandar fazer uma árvore com profundidade 1, 2, 3, ele vai fazer esta árvore aqui. Se a gente colocar a previsão, a predição do modelo, é a linha preta. Então, cada trecho desse aqui é uma folha, e cada trecho é uma folha da árvore. Cada trecho é uma folha da árvore. Então, essa árvore aqui não deve estar com a mesma profundidade da da esquerda, né? Porque ela deveria ter oito patamares e tem quatro só, né? Deve ser a árvore de profundidade 2 aqui. Essa que está na figurinha. Aí já dá para ter uma ideia boa do que o algoritmo está fazendo, né? Bom, como é que é a impureza? Vamos aprofundar um pouquinho mais. O erro é o valor do indivíduo real menos o valor predito. A soma dos erros... Por que a gente não usa a soma dos erros? Por que o diabo tem que usar esse quadrado aqui? Porque se a gente soma os erros, por construção isso aqui dá zero. Por quê, né? O y chapéu i, ele... Bom, se a gente somar tudo isso aqui, né? Os erros positivos. Os erros positivos vão se anular com os negativos. E, em geral, se a gente tentar minimizar isso aqui, vai dar zero, né? A gente consegue chegar em uma regra que os erros positivos compensam os negativos e vai dar zero. A soma dos quadrados dos erros, então, vai ser sempre um valor positivo. Então, agora, eu consigo pensar numa árvore tal que o yi fique mais perto possível do yi. Então, agora, eu consigo pensar numa árvore tal que o yi fique mais perto possível do yi. Então, agora, eu consigo pensar numa árvore tal que o yi fique mais perto possível do yi. Então, agora, eu consigo pensar numa árvore tal que o yi fique mais próximo possível do yi. Então, agora, eu consigo pensar numa árvore tal que o yi fique mais próximo possível do yi. Então, agora, eu consigo pensar numa árvore tal que o yi fique mais completo possível do yi. Esse algoritmo aqui, tipicamente, a gente chama de algoritmo de mínimos quadrados. Essa é a nossa impureza. Essa aqui é a nossa impureza. É isso que a gente vai tentar minimizar a cada passo. Quadrado médio do erro. O que ele vai ser? Ele vai ser a soma do quadrado. do erro dividido por N. Então, é isso aqui. É tipo o desvio padrão, só que em vez de ter o desvio em torno da média, é o desvio em torno da predição. Bom, e tem a ver com a variância? Ele tem, sim, tem. A variância do erro vai ser o erro... menos o erro predito, o erro médio ao quadrado, dividido por N-1. Erro predito, não, erro médio. O erro médio, em geral, é zero. Então, a gente pode fazer a soma dos erros ao quadrado. Desculpa atrapalhar. O senhor consegue ligar o pointer para melhor visualizar? Ah, consigo. Falta de lembrar só. Obrigado, professor. Então, a soma de quadrados do erro vai acabar sendo YI menos Y chapéu I ao quadrado, porque isso aqui é o erro, que tem tudo a ver. A gente está, de uma certa forma, diminuindo a variância do erro do nosso algoritmo. É uma leitura sobre o que a gente está fazendo. E como é que a gente avalia o modelo? A gente pode avaliar pela soma de quadrado do erro, pela média do erro, ou a gente pode avaliar pelo R². O que é o R²? A gente tem a variância do Y, para quem não conhece o R², que eu imagino que seja a maioria. Então, a variância do Y, a variância da minha variável resposta mesmo, sem modelo, a variância normal, é essa quantidade aqui. Soma de quadrado total, dividido por N-1. A variância do Y, é a variância do erro. É, aqui eu estou pegando a variância amostral mesmo, dividido por N-1, que é soma de Y em menos Y chapéu ao quadrado, sobre N-1. Está aqui. Soma de quadrado do erro é o que a gente quer otimizar. A gente vai comparar quão pequeno ficou esse cara, comparado com esse. Ou seja, quão pequeno é o erro do meu modelo, comparado com o erro que eu teria sem modelo nenhum. Chutando pela média, eu vou ter uma variabilidade, meu erro está sujeito a uma variabilidade, que é a variância do Y. É isso aqui. Chutando pelo modelo, eu estou sujeito a essa variabilidade de erro aqui. E daí, a gente vai comparar esse com esse. Então, o R² é 1 menos a variância do erro, dividido pela variância do Y. Então, o R² é 1 menos a variância do erro, dividido pela variância do Y. Então, o R² é 1 menos a variância do Y. Aí, o N-1 corta, no fundo é 1 menos soma de quadrado do erro, dividido pela soma de quadrado total. dividido pela soma de quadrado total. Isso é o R². Essa razão aqui é o erro, quanto que o erro representa da variação total. Tudo o que a gente não explicou da variabilidade do dado, 1 menos isso, é o percentual que a gente explicou da variabilidade do dado. dos dados. Por isso que o R² a gente pode dizer que é o quanto a variança foi reduzida por conta do modelo, ou o percentual da variança que o nosso modelo está explicando da minha variável target aqui. Então, vamos seguir aqui. Sem modelo, a gente está chutando pela média. E aí tem a minha arvorezinha aqui. Se a gente pegar a árvore e pegar o erro, a distância aqui deixa eu ver se eu consigo um rabiscadouro aqui. A distância da média até o ponto, a gente está medindo a variança por isso. O erro, a gente vai estar medindo a distância do modelo até cada ponto. Essas distâncias aqui. E aqui elas vão ser negativas. Aqui positivas, aqui negativas, aqui positivas e aqui negativas. E aí o gráfico de baixo é o erro. Exatamente essas distâncias. Por isso que o gráfico fica meio cortadinho aqui. A gente pegou esse trecho, centrou ele no zero, basicamente. Esse trecho aqui, centrou ele no zero. Aqui do lado também. Então, é isso que a gente está fazendo aqui. Assim que a gente explica o nosso modelo. Como é que eu volto aqui? Para o pointer. Ah, sim. Então, vamos lá. Daí, a variança do y. O total da variança, digamos, 30 e 2, 30 e algo. A variança do, do modelo, do erro da minha árvore, é esse tantinho aqui, uns 10. E a variança reduzida vai ser os outros 20 e poucos. Vai ser a parte laranjinha. O r² vai ser a variança reduzida dividido pela variança total. Isso é o r². Ficou aqui, né? Esse negócio. A gente apaga. Então, isso é o nosso r², né? Aí, a gente vai ter como estratégias de validação, a gente pode usar o r², a própria soma. A própria soma não tem uma interpretabilidade, né? Por isso que o r² acaba sendo bem popular. Mas, para uma mesma base, a soma é uma coisa que a gente compara quanto menor, melhor. O quadrado médio do erro é um pouquinho mais alto, né? Então, a gente vai ter que fazer um pouquinho mais interpretável. Mas, o r² é mais legal. E, uma avaliação gráfica do modelo é sempre interessante, sempre que possível, né? Assim, eu tive insights muito bons tentando visualizar o predito versus o observado do meu modelo ao longo de diferentes elementos no eixo x aqui, né? Vamos lá, então. Então, conceitos mais gerais. Desenho amostral. O que a gente vai falar de desenho amostral, validação cruzada e o tuning. O desenho amostral é um aspecto bem importante, assim, né? Como que a minha base foi gerada, tem um aspecto temporal entre as variáveis explicativas e a resposta. O que é a minha linha, afinal de contas, né? A minha unidade amostral é um indivíduo, é um grupo de indivíduos. É um paciente, é uma consulta médica do paciente, aí o paciente pode ter várias consultas, então a gente tem que dominar o desenho amostral, né? Aí a gente vai dar uma olhadinha como é que a gente faz o cross-validation disso e o tuning desse modelinho. Vamos lá para o nosso próximo notebook aqui. Vamos lá. É o aula 2, script 1, árvore de regressão, né? O outro era aula 1, né? Esse é aula 2. Aula 1 é porque o exercício era da aula passada e a gente está trazendo a solução. Agora é o da aula 2 mesmo. Vamos lá. Daí a gente importa os pacotes numpy, pandas, matplotlib, cborn. Ó, importante aqui, decision tree regress, é o regressor que vai fazer a árvore de regressão. Lembra do jargão, né? O modelo de regressão em machine learning case é que tem variável contínua. O modelo de classificação é que tem variável target categorizada. Então a gente vai rodar um decision tree regressor. Um plot tree aqui para a gente visualizar a nossa árvore. Trem test split. Outra coisa aqui que a gente vai fazer grid search cv. E o R quadrado aqui. Essa é a métrica. Scikit-learn, né? Tem um módulo inteiro de métricas. Dentre elas o R quadrado que a gente vai usar para avaliar a nossa árvore. Ó, essa primeira parte aqui a gente pode dar uma olhadinha, mas eu pretendo passar ela um pouquinho mais... Ai, esqueci de rodar a primeira célula. Rodem a primeira célula. E aí depois a segunda. Estamos definindo um random state. O linspace ele faz um uniforme entre esse valor e esse com mil valores igualmente espaçados. Então mil valores igualmente espaçados entre zero e um. Vai ser o meu x. ABC, os parâmetros da nossa parábola, que a gente vai fazer este bonito gráfico aqui, esse conjunto de dados. E a nossa... Espero que não tenha vazado o áudio aí. A nossa parábola aqui, tá? O parâmetro A que a gente definiu, o B, o vezes x e c vezes x ao quadrado mais um erro. Aí o erro eu coloquei uma... um truquezinho aqui para ele dar esses pontos mais afastados, esses outliers aí, né? E aí a gente joga tudo num data frame faz um gráfico e os elementos estéticos do gráfico. Aqui, tá? Então esse é o conjunto de dados que a gente vai tentar fitar. O procedimento é bem parecido. A estrutura é sempre a mesma. A gente vai definir a nossa árvore vai rodar um modelo com x e c. E com o y. E aí a gente vai aplicar o PREDICT, guardar os valores preditos e aí a gente vai dar uma olhadinha na previsão aqui, o que que tem nos valores preditos, ver que carinha que ele tem, calcular o erro aqui na mão, né? O algoritmo faz isso até, mas eu fiz questão de deixar na mão para a gente controlar tudo de uma... de uma forma mais explícita mesmo. Então, pode rodar esse aqui e a gente termina com um plot tree aqui, né? Que dá para a gente visualizar a nossa árvore. As regras são, se o x menor é igual a 851 vem para cá. A média do valor é 1,65. Então a média fica um pouquinho mais alta. Nesse caso aqui, se o x é menor que 0,176, vem para cá e a média cai para 0,77. Então, é uma coisa bem parecida, bem análoga ao que a gente fez na árvore de classificação. Só que a classificação é a média dos valores observados de cada árvore. Essa aqui é a média geral. Essa aqui vai ser a média de toda essa subpopulação aqui de 851 indivíduos, né? E essa é a média desses 176 indivíduos aqui. Bom, próximo gráfico aqui, para a gente visualizar o nosso algoritmo, o que a gente está plotando? x contra y vai ter um label de observado e a gente vai sobrepor com x contra p. O predito versus observado. Um dos gráficos mais básicos para a gente olhar um modelo. Isso aqui dá insights ótimos aqui. Por exemplo, a gente vê que tem bastante oportunidade de melhorar a árvore nesta região. Tem várias oportunidades aqui. O fit está bem grosseiro, mas já é bem melhor do que nada. Os próximos ganhos vão ficando sempre mais difíceis que os anteriores. Então, o ganho do modelo vai ficando uma coisa cada vez mais difícil. Próximo análise. A gente vai analisar os resíduos. Então, a gente vai fazer uma coisa parecida aqui. Aliás, agora a gente vai fazer um mosaico com duas linhas e uma coluna. O primeiro gráfico. Como é que eu sei qual é o primeiro gráfico? A x zero. Subplots me devolve uma figura e uma lista contendo os sistemas de eixos, que é o ax. Ax zero vai ser o gráfico da esquerda e x1 é o gráfico da direita. Aí a gente vai fazer no gráfico da esquerda os valores reais, os valores reais contra o predito. No gráfico da direita o resíduo da nossa árvore. Então, a gente rodando temos este bonito gráfico aqui. Que é bem o que a gente viu no PowerPoint. Aí dá para a gente enxergar melhor o modelo. Vamos para o próximo passo aqui. Aqui a gente vai fazer uma função um pouquinho mais complicada. Sim, complicada não. É tudo que a gente fez até aqui encapsulado. Vamos fazer a árvore com uma profundidade especificada no algoritmo. Vamos ter o ccp também. O ccp path. Mas estou usando a profundidade só para ilustração mesmo da técnica. A gente vai treinar o algoritmo. Calcular o predito. Calcular o erro. Plotar. E fazer o nosso gráfico. Esse gráfico aqui com a árvore com essa profundidade. Esse é exatamente o mesmo gráfico que a gente acabou de fazer. Pode rodar isso para definir a função. Na célula de baixo pode rodar também. A gente vai rodar a função. A gente está rodando para profundidades 1, 2, 3, 5, 10, 30. Vamos lá. Desde a 1. Profundidade 1. Ficou muito charmosa a árvore. Profundidade 1 é isso aí. Uma única quebra. Tosco. Vamos ver como a árvore vai melhorando aqui. Duas quebras. Melhor. Três quebras. Oito patamares. Já fica bem melhor. O curioso é que a árvore não pega uma simetria. Para começar a ficar simétrica tem que ter uma complexidade um pouquinho maior. Uma profundidade igual a 5. Melhorou ainda o algoritmo. E se a gente aumentar mais profundidade 10, ela já dá umas escapadonas. Pega um único ponto lá em cima. E aí com profundidade 30 ela está literalmente passando em cima de cada ponto. 30 é muito de profundidade. Ela vai ter tranquilamente uma folha para cada elemento. E o resíduo vai ficar zero. Esse é um bom algoritmo? Olha, eu diria que até é melhor do que nada. Mas é o melhor algoritmo? Aí talvez não, porque se esses erros são aleatórios e imprevisíveis, talvez fosse melhor uma coisa como isso, mais suave. Esse aqui ainda tem umas coisinhas ali que talvez a gente consiga limpar com o CCP Path. É... E... Aí depois a gente vai fazer um Grid Search. Mas deixa eu voltar para o slide. Depois a gente... Vamos falar primeiro o que é o Grid Search. E aí depois eu explico. Depois a gente volta lá no algoritmo. Vamos voltar para o slide aqui. Ju, tem alguma pergunta aí? Alguma questão que vale a pena a gente responder? Temos, temos sim, professor. Vou trazer para você. Bom, o João, lá de Sabará, Minas Gerais, ele perguntou se o senhor poderia dar uma refinada na explicação dos gráficos. E se é possível também, professor, aumentar um pouquinho a tela para eles enxergarem melhor. Sim, aumento. Faço as duas coisas. Vamos lá. Deixa eu ver se eu boto uma lupa aqui. Então, esse primeiro gráfico aqui, ele é o valor... É a minha target mesmo, o Y que eu estou tentando prever. E o X que é a minha variable feature. Poderia ser, por exemplo, renda da pessoa e tempo de profissão. Aí, sei lá, a gente poderia observar um padrão que quanto mais tempo de profissão, maior a renda, a partir de um certo ponto... Bom, tomara que isso seja mentira. Para mim, daqui para frente é só para trás. Mas, a bolinha são os valores observados. Cada bolinha é uma linha da minha tabela. E a linha é o predito. Então, a gente consegue ver o que o modelo estava falando versus o que aconteceu de verdade. Do lado esquerdo é o resíduo. O resíduo já é a diferença entre a bolinha e o tracinho vermelho. Então, a gente vê aqui padrões que são recortes mesmo do nosso dado original. Então, daqui até aqui, esse patamar, esse platô, ele é bem visível aqui. Do 0,3 praticamente até o 0,7 um pouquinho antes. Aqui, do 0,3 até o 0,7 a gente está nesse trecho aqui. O resíduo vai ser cada pontinho menos o valor dessa linha. E aí, assim por diante. O resíduo fica com essa cara aqui. Bom, é isso que significa esse gráfico aqui. Tem mais algumas? Tem sim. Pronto. Vamos lá. O Valdely... Não, desculpa. Vamos lá. Aqui... Tem uma da Argentina. Deixa eu achar aqui. O Valdely, no Lange, lá da Argentina. Ele respondeu assim. Professor, consegui um coeficiente de Gini de aproximadamente 50% e 50% para treino e teste. Isso. Acho que foi mais ou menos o que a gente conseguiu aqui também. Muito bom. 50% em geral é um coeficiente de Gini respeitável. Claro que depende do problema. A gente pega um problema muito simples, por exemplo, descrever o movimento de uma bolinha que a gente solta ela de um patamar. A gente registra a posição dela e faz um modelo para isso. O modelo tem que conseguir uma previsão muito maior. Essas coisas físicas, mecânicas, em geral a gente consegue precisões muito boas. Agora, essas coisas mais comportamentais, ciência social aplicada, os erros são bem com outra cara. São grandes padrões que a gente acaba pegando mesmo. O que mais? Por enquanto são essas, professor. Então, vamos lá. Dá tempo de a gente andar mais uns 5 minutinhos aqui e o recreio é 9 e 10. 8 e 10. Mas se precisar, a gente ajuda. Tá bom. Vamos lá. Preciso trocar essa imagem aqui. Dá um pouco de aflição essas imagens geradas por IA. Enfim, a ideia ficou fofinha. Mas, vamos lá. Quais são os objetivos segundo Tibichirani? Os objetivos da validação cruzada. O primeiro objetivo, que é o que a gente vai explorar, até agora, é fazer o tuning do modelo. Achar qual é o melhor conjunto de hiperparâmetros para aquele algoritmo. Segundo objetivo é ter uma expectativa mais acurada da qualidade do nosso modelo. Por exemplo, qual deve ser o AUC quando eu botar o meu algoritmo no valendo. Na realidade para ele tomar decisões. Então, a ideia básica. Bom, só para reforçar. A gente está aqui do lado esquerdo. A gente vai falar de tunar o modelo. Basicamente. Então, a ideia básica. A gente vai dividir a base em treino e teste. Rodar o modelo com uma configuração de hiperparâmetros. Avaliar o modelo na base de teste. Repetir os passos 2 e 3. Isso é o que a gente fez para todas as configurações diferentes de hiperparâmetros. Esse algoritmo básico que a gente fez até agora. Resgatando um pouco. A gente tem uma acurácia. Uma qualidade do modelo. De novo, não é assertividade. Assertividade é fazer uma afirmação de forma enfática. É acurácia. A gente roda o modelo na base de treino. Avalia na base de teste. É um modelo complexo. Tende a ter uma acurácia excelente na base de treino. E ruim na base de teste. Aí tem um modelo um pouco mais equilibrado. Em geral é comum cair um pouquinho a acurácia. E um modelo bem mais simples que no exemplo aqui. Um modelo mais simples acabou sendo o melhor aqui na ilustração. Qual que é o melhor de todos? Depende. Depende do banco de dados. A gente faz isso por tentativa e erro. É uma força bruta mesmo. A gente vai testando várias configurações de modelo até achar uma que seja a melhor de todas. Pelo menos as que a gente testou. A gente vai para uma terceira ideia. Uma segunda ideia, aliás. Que é dividir a base em treino, validação e teste. Para que serve o treino? Para a gente criar as regras. Validação. Para tunar o modelo, a gente escolhe a melhor configuração de hiperparâmetros aqui. E aí depois a gente treina o melhor modelo com tudo isso aqui. E avalia na base de teste. Essa é a cartilha do Tibichirani. A gente avalia o modelo finalmente na base de teste. Então, é o Kfold. É uma terceira ideia. O Kfold eu vou tomar a liberdade aqui de abrir uma animação que eu fiz. Deixa eu abrir ela aqui. Bom, isso tudo são coisas que a gente já passou. Eu vou para a parte do Kfold logo. Isso aqui a gente já falou. Vai filho. Essa animação eu devia ter tirado mesmo. Eita. Deixa ele. Vai ficar fazendo a validação aqui. Bom, vamos ficar aqui mesmo. O Kfold, como o próprio nome diz, a gente divide em K grupos a nossa base de treino. No caso eu tenho um, dois, três, quatro grupos. Que a gente chama de Folds. Então eu tenho K grupinhos. E daí a gente vai desenvolver um modelo. A gente separa um desses grupinhos para teste, para validação. Treina um modelo com todos os outros e valida aqui nesse grupinho. E aí depois a gente vai revezando qual que é o grupinho. Aí depois a gente separa esse grupinho para validação. Treina eles com um, dois três, quatro. E avalia o modelo nessa base de teste aqui. Depois faz o mesmo com o segundo grupinho. Tira ele para validação. Treina com o três, quatro. Avalia no dois. Depois treina o modelo com o dois, três, quatro. E avalia no um. Aí a gente calcula a média das acurácias nas bases de validação cruzada. A gente avalia sempre na validação cruzada. Por que tudo isso? Se a gente faz dessa forma treino, validação e teste a gente pode falar poxa, rodei o modelo de novo eu fiz uma outra amostragem deu um resultado diferente. E agora? Independente se é melhor ou pior é diferente. Como é que eu garanto que o meu modelo não ficou fortuitamente um modelo sub ótimo? Essa ideia do CAFOLD é uma ideia para a gente ela tem alguns conceitos legais. Uma, que a gente consegue eventualmente uma base de validação um pouquinho menor até. Porque a gente acaba fazendo a validação com toda a base de dados. E aí se a validação é menor o treino é maior também. Outra ideia legal, a gente faz repetições da validação cruzada. Então a gente consegue afirmar com uma segurança muito maior que aquele conjunto de hiperparâmetros é melhor. E tem até uma técnica chamada INCAFOLD que está mais ali na sequência. A gente faz várias vezes o CAFOLD. Qual que é a ideia? A gente vai fazer quatro vezes o modelo para cada combinação de hiperparâmetros. Então a gente vai treinar o modelo 1 uma, duas, três, quatro vezes separando o primeiro fold para teste, separando o segundo para teste, separando o terceiro para teste, separando o quarto para teste. E aí a gente faz uma média dos indicadores. Aí a mesma coisa para o modelo 2 a mesma para o 3. E aí o modelo escolhido tipicamente é o modelo que tem maior acurácia. Este é o CAFOLD. Deixa eu ver se a minha ... Estava no negócio errando aqui. Esse que era o meu CAFOLD. Tudo bem, eu vou deixar o link para vocês. Tem uma animação bonitinha aqui do CAFOLD. Vou deixar no chat aqui do Zoom. Depois vocês disponibilizam para a turma por favor. Talvez não abra agora, mas eu preciso deixar público aqui para vocês poderem ver. Bom, a gente já falou desse dessa ideia de usar o CCP alfas que é uma forma de tunar a árvore por exemplo. No nosso código a gente vai usar essa função GRID SEARCH CV e tem esse parâmetro CV igual a 4 a gente está especificando um CAFOLD com 4 etapas. Tem mais detalhes aqui, mas acho que está na hora do recreio. Vamos para o recreio, Ju? E aí a gente volta na sequência aqui explorando o GRID SEARCH CV no Python. A gente volta lá no nosso código e roda ele. É isso aí, pessoal. Então vamos ao intervalo e na sequência a gente volta com a rodada de perguntas. Até já, já!

BLOCO 2

Pessoal, vamos voltando e vamos voltando com uma rodada de perguntas para o professor Juca. Então, vamos lá. Professor, o Rodrigo, lá de Pedra Preta do Mato Grosso, Rodrigo, um abraço para você, viu? Perguntou o seguinte. Professor, sobre o exercício, ao rodar o treino da árvore, observei que há um desbalanceamento em relação aos dados-alvo. Ou seja, em torno de 75% dos dados indicam a não inadimplência. O algoritmo de árvores é sensível a esse desbalanceamento? Esse evento pode causar um certo... Vies de padrão? Professor? Rodrigo, obrigado pela pergunta. A árvore vai bem com dados desbalanceados. Se o desbalanceamento é muito intenso, aí tem técnicas que fazem a árvore ficar um pouco melhor. Tipicamente, detecção de fraude tem um evento super raro, ainda bem. E... Aí o algoritmo tem... Bom, é um mundo que escapa um pouco do nosso escopo aqui, né? A ideia é a gente ter um básico para a gente conseguir expandir o nosso conhecimento na direção que for mais conveniente para a nossa pesquisa, né? Aí já começa a ficar um pouco específico, né? Como série temporal, assim. A gente não vai falar de série temporal com relação ao serial, assim. Dá para a gente colocar tudo isso na nossa temática, mas não é nosso escopo. Hoje. Tem mais alguma? Mais uma, né? Sim, sim. Vamos lá para a prova. Pode se mandar. Deixa eu trazer aqui para você. Temos a do Anderson. Anderson lá do Rio de Janeiro. O Anderson pergunta o seguinte, professor. Como outras métricas, como precisar... Como precisar... Como precisar... Como precisar... Como precisar... Como precisar... Como precisar... Como precisar... Como precisar... Como precisar... O F1 score poderiam complementar a análise. Tá. Essa, eu acho que ele está falando do exercício mesmo, né? Bom, F1 score, precision e recall são uma ideia bem análoga com a especificidade e sensibilidade, mas ela é meio que transposta, né? Em vez de condicionar na linha, a gente condiciona na coluna e vice-versa. É uma ideia bem análoga, assim, né? Enfim, são outros indicadores. Aí eu tenho que escolher alguns para colocar aqui, porque senão vai ficando muita coisa e a gente não consegue dar conta. Mas, assim, em alguns problemas específicos, o pessoal olha, sei lá, a... Dá um pouco mais de peso. Dá um pouco mais de peso para a sensibilidade, por exemplo, né? Enfim, tem alguns problemas específicos, você pode ter alguma particularidade que para ele um determinado indicador é melhor. Assim, eu estou falando de indicadores bem gerais, super populares, amplamente aceitos, né? E daí a gente tem condições de ampliar o nosso próprio... Aprendizado, né? Então, mas é uma ideia bem análoga ao precision... A sensibilidade e especificidade. Bom, vamos lá, seguindo aqui, né? A gente vai agora fazer o nosso grid search. Para o grid search, a gente vai fazer, primeiro, estabelecer os parâmetros do grid. Parâmetros do grid, a gente estabelece condicionados. Condicionado, que vai dar o nome do indicador e uma lista com as possibilidades que a gente vai testar, para qualquer, todos os indicadores que a gente quiser. Por exemplo, max features aqui, é um parâmetro desse algoritmo aqui, que é o random forest, que eu estou testando ele nesse range de 1 a 11. Assim, né? Esses parâmetros são do algoritmo que eu vou usar, o RF model, que eu estou... Estabelecendo aqui. Eles têm que ser parâmetros aceitos desse algoritmo aqui. E aí, eu vou chamar o grid search assim. Grid search, igual ao grid search cd, essa funçãozinha que a gente importou. O estimador é o modelo que eu estou usando. Pode ser uma árvore de classificação, random forest, é indiferente. É indiferente, assim, tem que estar tudo consistente, né? Os parâmetros têm que conversar com o modelo que eu colocar aqui. Mas, assim... O scikit-learn aceita vários modelos diferentes, uma classe grande de modelos. O parâmetro grid são os parâmetros do grid que eu defini no dicionário. O scoring é o AUC rock, é o que ele vai usar para tunar, escolher o melhor modelo. Ele vai fazer um kfold com 4 folds, né? 4 dobras. E any jobs igual a 1 parâmetro para ele, se tiver... Vários cores no processador, ele usar todos eles, né? É um parâmetro para pegar o máximo possível do processador. Bom, kfold, né? A gente... Resgatando a ideia um pouquinho, né? O kfold, a gente separa uma base de validação, treina com o resto, valida. Aí, separa o outro fold, treina com o resto, valida o fold que separou. E aí, assim por diante para todos os folds. O nkfold é uma estratégia mais limite ainda, né? Ele faz o kfold uma vez, permuta os dados e faz outro kfold. Embaralha os dados, faz outro n vezes. Assim, a gente vai aumentando bizarramente a complexidade do algoritmo. Cada coisa dessa é multiplicativa. Dá pra... Dá pra aumentar bem a complexidade. O algoritmo, felizmente, ele roda até bem rápido, né? Assim, a gente processando bases de dados, assim, numa memória RAM do computador, ele acessa o dado super rápido, ele roda super rápido, as operações não são demoradas. Mas se a gente começa a fazer muitas e muitas vezes, aí pode começar a pesar, né? O nkfold... O nkfold, a gente tem que ter máquina, tem que ter tempo, tem que ter... Tem que ter alguns elementos aí pra fazer o nkfold. Mas é uma ideia interessante, né? Deixa o computador rodando na sexta-feira, pega ele pronto na segunda, depois de um final de semana. Bom, é isso. Vamos lá pra parte 2 da nossa árvore de regressão. A gente vai fazer... O nosso kfold aqui. Ah, eu vi que tinha alguma pergunta sobre o gráfico, né? Deixa eu tentar dar uma repassadinha aqui no gráfico que a gente... Eu acho que o professor deu uma travadinha, pessoal. Então, vamos aguardar. A gente vai aguardar que ele já está restabelecendo o sinal. Vou dar um tempinho pra ele. E se vocês vierem a ter quaisquer dúvidas, pessoal, enviem na aba de perguntas, tá bem? Porque os professores vão selecionando e nós vamos transmitindo ao professor Juca. Tá bem? Às vezes a gente não consegue levar todas as perguntas devido à demanda de conteúdos da aula. Tá bom? Mas a gente vai ajustando aqui o máximo que a gente puder, tá bom? Dá um pouquinho, professor. Já voltou. Pessoal, eu quero saber quem de vocês está acompanhando as defesas de DSA. Elas iniciaram no dia 13, certo? E elas vão até o dia 24. E eu vou pedir aqui pras meninas colocarem pra vocês o link das defesas que vocês podem e devem, tá bem? Assistir. E vou aproveitar, enquanto o professor restabelece a conexão dele, pra passar um recadinho pra vocês. Então, ó, turma 232 e turma 241, né, estão sendo, né, liberados aí o podcast do TCC pra vocês, certo? E também foi encaminhada uma pesquisa pra que a gente possa entender melhor, tá bom? A necessidade que vocês têm pra aprimorar o podcast. Então, por favor, recebam. Respondam lá pra nossa equipe, tá? Turma 232, ó, recadinho importante pra vocês. Do dia 30 de janeiro, que virá, certo? Então, a partir do dia 30 de janeiro até o dia 6 de fevereiro, é a entrega dos resultados preliminares, certo? A respeito do TCC. Então, é importante a manutenção do diálogo de vocês, juntinho, a equipe, tá bom? Ao orientador. E também vocês podem contatar a equipe em qualquer dificuldade que vocês tenham. Então, turma 232, do dia 30 ao dia 6 de fevereiro, tá? Entrega dos resultados preliminares do TCC. Mantenha um bom contato com o orientador. E qualquer dúvida, dificuldade, pode contatar a nossa equipe. Turma 241. Estamos realizando o processo. O processo de designação dos orientadores de vocês. E na primeira quinzena de fevereiro, né, terá esse alinhamento entre alunos e orientadores, tá bom? Então, na primeira quinzena de fevereiro, já deve acontecer aí este alinhamento. E, turma 242, a partir de fevereiro, sim, o podcast TCC estará liberado pra vocês. Bom, então, fique. Fiquem atentos aos recadinhos que nós trazemos aqui, certo? E qualquer dúvida que vocês tenham, vocês sempre podem mandar pra nós e nós auxiliaremos da melhor forma. E olha, pessoal, mantenha orientações de tema de podcast, porque a finalidade do podcast é justamente pra auxiliar vocês, tá bem? Da melhor forma, né? É um plus aí que nós temos. Pra auxiliá-los. Aliás, vão mandando aqui pra mim, aproveitar enquanto o professor está reiniciando a máquina dele, qual o TCC, qual o tema que vocês, né, acompanharam, que vocês gostaram bastante, né? Que vocês sentiram aí. Passem pra nós, por favor, tá bom? Vamos lá, o Robson perguntou aqui. O tema do TCC é livre ou o orientador que indica o tema? É escrita, monografia ou artigo? Você vai receber, Robson, as coordenadas do seu orientador, tá? Vocês recebem direitinho, né, o script. O tema, procure fazer, né, aquele que é mais indicado, que você se sente mais confortável, né, dentro de todo o conteúdo que foi visto do curso. Mas converse sempre com o orientador. Aliás, gente, o orientador, ele tem um grande papel aí. Ele vem pra auxiliar, pra facilitar todo esse processo pra vocês, tá bem? Bom, o professor está voltando e eu vou passar aí a palavra pra ele. Professor? Oi. Ai, gente, nossa, sabe que eu tenho pesadelo com isso, de dar tela azul no meio da minha aula. Foi o que aconteceu. Bom, e aí o computador fica atualizando. Vamos lá, que a gente tem bastante coisa aqui. Deixa eu retomar aqui o nosso... Nosso... Spider, né? Então, eu estava comentando como é que é o gráfico, né? Assim, eu até estava abrindo a minha lozinha. Foi exatamente quando deu a tela azul. Então, só vou pedir licença pra não abri-la agora. Oi. Enquanto o senhor estava reiniciando aí a tela azul, o Gustavo, lá do Rio de Janeiro, perguntou se a sua camiseta é do jogo Metroid. Não, não é. Eu adoro o Metroid, mas é um homem de ferro aqui, ó. Passando roupa, entendeu? Homem de ferro. Bom, a gente estava falando... Que tipo de insight que a gente quer com esse gráfico. Eu quero ver, simplesmente, se o observado está perto do esperado. Ah, ficaria até legal que a gente compor isso com um R² aqui, né? Uma medidinha aqui, mas eu não deixei no gráfico, não. Mas dá pra colocar, não é? Calcular o R², aliás, é bem fácil. Colocar no gráfico, nessa era de chat GPT, não é uma coisa muito difícil, não. Vamos lá. É frustrante quando você não sabe o que tem que fazer, mas você tem o gráfico. Falar pra ele, ó, bota o R² aí, aí ele dá ajuda. Vamos lá. Então, antes que a gente estava aqui, no... Ah, eu vou ter que rodar isso aqui tudo de novo, né? Peraí, vamos lá. Pra gente retomar da mesma... Carregamos as bibliotecas, geramos dados, nossa árvore, a primeira árvore, então, aquele monte de árvore, e... Aí aqui a gente separa em treino e teste. A gente vai fazer um K-fold com o holdout. É aqui, nesse ponto que a gente parou. Pode rodar essa célula. Essa base de teste a gente vai usar pra avaliar a curácia, pra ter uma estimativa mais confiável da curácia quando o modelo estiver no valendo, né? Então, a gente vai fazer o cross-validation na base de treino. A gente vai fazer o K-fold na base de treino, né? Então, a gente vai, no fundo, usar todas as observações pra treinar o modelo e avaliar o modelo em todas as observações, de alguma forma. É o que o K-fold faz. Uma metodologia muito legal. Próxima célula aqui. A gente vai... É... É... Definir a nossa... A gente vai usar a árvore. A gente tá até fazendo um fit aqui. E... É... Por que esse fit? Aqui a gente tá deixando a árvore ser feliz. Né? Aí a gente fita ela com o modelo e a gente vai pegar o custo de complexidade. É... Assim, né? Qual é que é a fórmula do custo de complexidade? Na verdade, eu nem sei explicar, assim, de bate-pronto, assim. Mas o conceito é o importante. O conceito é que o custo de complexidade é um parâmetro que, se ele é pequenininho, a árvore não vai ter uma penalização na função objetivo dela, na soma dos quadrados, por causa da complexidade, então ela vai tender a ficar gigante. Se a gente coloca um parâmetro alto pro custo de complexidade, ele vai podar mais a árvore e a árvore vai ficar... É... É... Menorzinha. Tá? Então a gente vai pegar todas as possibilidades do custo de complexidade para este conjunto de dados aqui, com essa topologia de modelo. E... Aí a gente vai... É... Fazer isso aqui. A gente vai... É... É... É... Deixa eu ver aqui. A gente tá calculando... Bom, tá repetindo a... Essa instrução aqui. Estamos definindo o path, que é... O custo de complexidade, o CCP alfas e a impureza. A gente vai pegar o CCP alfa do path e a impureza. A impureza a gente não vai usar. A gente tá mais interessado no CCP alfa, mas dá pra gente fazer um gráfico aqui bonitinho do... Bonitinho, não. Esse aqui tá meio feioso, né? Mas mais pra gente ver como é que é a evolução da impureza de acordo com o custo de complexidade. A impureza tá crescendo com o custo de complexidade, porque esta é a base de treino. Se fosse a base de teste, a gente ia olhar uma coisa e ia ver um comportamento bastante diferente. Tá? Tudo bem? Então, assim, o nosso grid a gente definiu pegando os custos de complexidade. O... Nosso parâmetro de grade, vai ser... Como tem muitos CCP alfas, eu tô pegando eles de 10 em 10 pra não... Pra gente fazer uma rodada mais pragmática aqui, né? Então, esse vai ser o nosso grid. O grid search CV é o único parâmetro que a gente vai mexer. A gente vai pegar a árvore. O parâmetro de árvore que a gente definiu lá em cima, né? O parâmetro de árvore que a gente definiu lá em cima, né? Com esse grid, com um k-fold de 5. E a gente vai otimizar a média negativa do quadrado do erro. Por que negativo? Pra ele... É um truque que o pessoal usa em computação, né? Eles fazem, às vezes, um algoritmo que maximiza uma função objetivo. E aí, se eu tenho um erro, eu pego o erro negativo e maximizo ele e dá certo também. Mas é a mesma coisa que a soma dos quadrados do erro. A gente tá fazendo a mesma coisa aqui. Reduzir a soma... É um algoritmo de mínimos quadrados também. A gente vai fitar aqui a nossa árvore. Vamos rodar todas essas células aqui. Essa aqui a gente já rodou. Essa aqui também. E essa de baixo aqui. Que a gente... E, ó, o gridSearch, ele vai rodar alguma coisa nessa instrução aqui. Até aqui, ele tá só definindo as propriedades do algoritmo, os hiperparâmetros. Aí é que a gente roda. E no nosso objeto tree, esse objeto tree tem todos os atributos e métodos que a gente vai precisar. Esse objeto gridSearch também. E tem mais algumas coisas interessantes. Quando a gente fita, ele guarda um metadados de todos os modelos que a gente usou. Então, olha só. A gente pode olhar, por exemplo, ele tem um atributo chamado bestParams. Vamos ver o que tem nele. Tem os parâmetros, que no caso é um só. O parâmetro que otimizou a nossa árvore. Que é esse custo de complexidade aqui. E aí a gente pode usar ele pra treinar o modelo final na base final, como recomendo o Tibichirane. Mas ele até já faz isso aqui dentro pra gente, né? O que mais que a gente tem aqui? Vamos... Vamos rodar agora o modelo com a melhor combinação... Com a melhor combinação de hiperparâmetros. A gente pode rodar desse jeito aqui, né? Quando a gente coloca dentro do método, a gente tá rodando bestTree. Vai ser o decisionTreeRegressor, asterisco, asterisco, bestParams. É o jeito que a gente fala pra ele, pra ele pegar esse dicionário, né? Que é o bestParams. E ele vai definir o parâmetro ccpAlpha com esse valor aqui do dicionário. Isso aqui seria equivalente a gente... A gente escreveu isso aqui, ó. ccpAlpha igual... Ah, e aquele número ali, 0.00013... Bom, e aí por aí vai, né? Então eu vou deixar como tava aqui, porque é equivalente, mesma coisa. Então a gente roda esse carinha aqui. E temos a nossa árvore fitada aqui nesse objetinho. É... bestTree. Aí a gente faz a predição, calcula o R². Olha o R² quanto deu aqui. 96%. É um R² bem bom aqui já, né? Na base de teste ainda. Então, é... Conseguimos fazer um modelo bem razoável aí, bem interessante. E aí... Ah, esse último gráfico ele foge, tá fugindo um pouquinho do padrão, né? Porque é uma... É uma análise que eu acho interessante da gente fazer. O padrão de análise que a gente tem feito até agora, né? É... Aqui, a ideia é ver os valores reais contra os valores preditos. A gente espera que os pontos estejam perto da reta de 45 graus aqui. Bom, só não é 45 porque não tá em escala, né? Ele tá um pouquinho mais largo do que alto. Mas a gente espera... Ver os valores esperados perto do observado. É... Um gráfico interessante, uma ideia de análise interessante. Dá uma ideia pra gente também de... É... Quanto que a gente consegue discriminar de perfis diferentes, né? Se for... A gente tá tentando... Não sei... É... Prever... Uma medida... É... É... Médica... Que se obtém de uma forma... De uma forma invasiva, né? No paciente. Mas... A gente usou um modelinho com outros parâmetros não invasivos. É... Quão bom é o nosso algoritmo, né? Aqui dá pra gente ter uma ideia visual, assim, do erro, né? Um pouco além do R². Dá pra gente ver se tá errando mais pra valores altos ou mais pra valores baixos. Esse tipo de insight que a gente vai recolher aqui. Tá bom? Então... É... Vamos voltar pra... A nossa... Apresentação. É... A gente vai falar de Ensemble agora, né? O que que são... É... Modelos de Ensemble. É... Onde vivem. O que comem. E... Os predadores naturais dele, né? É... O que que eles são? Assim... Os modelos de Ensemble fazem exatamente a mesma coisa que os modelos de árvores. A árvore que a gente viu, né? Eles têm... É a mesma... É... É... É... Aplicação. A mesma... Categoria de modelos. Que... Qual que é a ideia de um Ensemble? Ele é uma mistura mesmo de modelos. Eu vou pegar modelos diferentes, de algoritmos diferentes ou não. E eu vou misturar. Essa é a ideia básica. É... Bom... Sempre bom a gente lembrar do Crispy, né? A gente não fala muito de business. Understanding realmente não é o nosso escopo. Data Understanding já teve, né? Algumas matérias de exploração de dados, tratamento, tarará, preparação dos dados. A modelagem é o que a gente tá falando. Né? E aí, depois a gente vai avaliar o modelo e implantar. A gente vai precisar fazer tudo isso. O Ensemble é... É... Assim, ele é um modelo... O que a gente vai ver, pelo menos, é um modelo supervisionado. Dá pra gente fazer Ensemble também. De modelos não supervisionados. Mas a gente vai discutir esse lado esquerdo aqui. É... A gente pode usar tanto pra algoritmos de regressão quanto classificação. É... E a gente tá, de fato, num paradigma de um Machine Learning, né? É... Embora a gente pode... É... Fazer o exercício de pensar numa ideia de um modelo estatístico com Ensemble também. É... Tipos de Ensemble famosos, né? O Bagging, o Boosting e o Stacking. As três grandes classes de Ensemble. O que que é o Bagging? A ideia do Bagging... A ideia é a gente... É... Pegar vários modelos e pegar a média das previsões. Ou como um... Uma votação dos modelos, né? Um modelo de classificação. A classe que for mais votada pelos diferentes modelos é aquela que a gente classifica. Essa é uma ideia de Bagging. Eu faço vários modelos diferentes, pego a média da probabilidade predita ou do valor predito dele. Outra ideia de Ensemble. A ideia do Boosting é a gente tá sempre tentando melhorar o Evo do modelo anterior. É isso que faz um Boosting. O Extreme Gradient Boosting é um tipo específico. É de Boosting. E o Stacking. Stacking já é uma coisa diferente. A gente pega a saída de vários modelos e coloca como entrada dos outros modelos. As Features passam a ser modelos. Essa é a ideia de um Stacking. Bom, é bastante usado, por exemplo, quando alguém compra um modelo desenvolvido por terceiro. Tipo, Serasa, Boa Vida. Vista, esses birôs aí. Vendem modelos, não só de crédito até. E aí as pessoas pegam esses modelos e fazem um Stacking nos próprios modelos. É um jeito que se faz bastante. Vamos entrar no Bagging. O que é um Bagging? Bagging é um Aggregation. É um tipo de Aggregation. O que é um Aggregation, no final das contas? Bom, vamos imaginar aqui três modelos. Três classificadores. É uma varinha mágica aqui, sem imaginar que modelo que é. Mas pode ser uma árvore, por exemplo. Mas três modelos. O objetivo é a gente usar a sabedoria desses três modelos. A ideia mais básica é fazer um voting. O primeiro modelo falou que é azul. O segundo falou que é vermelho. O terceiro falou que é azul. Classificamos como azul, mais votado. Uma outra ideia é... O primeiro modelo falou que a probabilidade de azul era 3%. O segundo falou que era 7%. O terceiro falou que era 20%. Tira a média. 4% vai ser o palpite do meu novo modelo. E aí eu uso para classificar esse 4% do jeito que eu quiser. Essa pessoinha aqui. A pessoinha está aqui porque... Imagina que eu vou classificar esta pessoinha. E aí 3% da pessoinha será... Classe azul. E agora a gente vai querer saber... O que a gente precisa? Ter modelos diferentes. Que mirem no mesmo alvo. Na mesma target. Que sejam úteis, bons o suficiente. Não adianta o modelo escapar muito do target. E eu preciso que eles tenham diversidade. Eu preciso que eles sejam diferentes. Então... Tem uma história do... Esquecer um filósofo. Esqueci o nome do cara. Mas ele disse que foi numa feira que tinha que chutar o peso de um boi. Aí ele não tinha a menor ideia quanto devia pesar um boi. Aí ele tirou a média de todos os palpites. E deu o palpite dele. Diz a lenda que ele ganhou. Enfim. E aí... A gente... Traz essa ideia... De uma sabedoria de multidões. Um modelo pode errar... A ideia é que ele vai errar pra cima uma observação. Outro vai errar pra baixo. Outro vai errar pra cima. E aí acaba... Um erro compensando o outro. E a gente tendo uma curácia melhor. Essa... Esse é o princípio. Ou pelo menos o desejo. Vamos fazer uma ilustração. Se eu tiver vários modelos diferentes. Tirando na mesma target. A média de todos eles talvez fique melhor do que cada um deles individualmente. Então como é que eu gero diversidade nos meus modelos. Com a qualidade aceitável. É... Bom. A gente pode fazer modelos diferentes de alguma forma. Os algoritmos diferentes. Os hiperparâmetros diferentes. Mas... Talvez não seja uma boa ideia. Ao invés da gente tentar mexer no algoritmo. O que se a gente mexer na base? É... Então assim... Se a gente... Permutar as linhas da base. Sortear elas de alguma forma. E fazer um modelo diferente. Com variações da mesma base. É... Bom. Aí a gente vai entrar na ideia do bootstrapping. O que é o bootstrapping? Bom. Bootstrapping é um algoritmo computacional. Super famoso. Vamos dar uma entendida aqui. O que é o bootstrapping? A gente volta no bagging já. Vamos imaginar que eu não sei calcular. Eu sei calcular a média. Eu tenho uma média do saldo dos contratos da minha base aqui. Por algum motivo eu preciso de um desvio padrão dessa média. Eu tenho uma média amostral de alguma coisa importante para mim. Satisfação do cliente. Sei lá. E... É... É... É... É... É... É... É... É... Eu quero ter uma ideia do desvio padrão da média. Não é o desvio padrão do x. É da média. Qual que é o exercício? Eu tenho a base. Calculei a média. Se eu tivesse um monte de outras bases do mesmo tamanho. Coletadas com um processo igual. Qual que seria a variabilidade? E calculasse a média para cada uma dessas bases. Qual que seria a variabilidade da minha média? Ela está sujeita a que tipo de erro? Suponha que eu não sei calcular a média. A gente pode calcular. A gente pode calcular por bootstrapping com a seguinte lógica. É... A gente... É... É... É... Sorteia... É... Com reposição. Eu tenho n indivíduos. Digamos 20. Eu sorteio 20 indivíduos com reposição. Calculo a média. Essa é uma amostra bootstrap. A primeira. Aí eu sorteio de novo com reposição. Aí aqui no caso, ó... Peguei o indivíduo amarelo. O vermelhinho. O amarelo de novo. não tem problema, o azulzinho e o verdinho. Aqui eu peguei o cinzinho que não apareceu aqui em cima, o azulzinho, o verdinho e o amarelinho duas vezes. Aqui eu peguei o cinzinho, verdinho, cinzinho, verdinho, cinzinho, não tem problema. Repetiu, repete mesmo, é com reposição. A gente calcula a média em cada uma dessas vezes e aí calcula o desvio padrão. Então, essa é a ideia do bootstrapping. A gente tem um conjunto de dados de tamanho N, a gente quer estimar um parâmetro por bootstrapping, o erro padrão, os estatísticos usam muito isso, às vezes é difícil estimar um erro padrão, lá eles fazem por bootstrapping. Aí a gente retira uma amostra aleatória de tamanho N, calcula o parâmetro e armazena. A gente vai fazendo isso várias vezes e depois avalia. A estimativa do nosso parâmetro, a distribuição das amostras bootstrapping dele. A gente pode repetir isso, digamos, 10 mil vezes, bastante mesmo. E aí a gente calcula o nosso parâmetro do jeito que a gente quiser. Esse exemplinho está implementado aqui no próximo código, que é o bootstrapping. É um código mais curtinho, que a gente está usando num Py, Pandas e Matplotlib. Pode abrir ele, o código aula202bootstrapping.py. Pode abrir comigo, pode rodar a primeira célula. A segunda, a gente está gerando um dataframe. Um dataframe DF. Vamos dar uma olhada nela, o que que tem nesse dataframe aqui. Cadê ele? Aqui, ó. Ele tem uma coluninha só, chamada dados. Tá bom? Isso é o nosso dataframe. E aí a gente vai fazer uma funçãozinha aqui para calcular o bootstrap. A gente vai inserir os dados. O número de amostras bootstrap. É... As médias. Eu vou definir como um array do numpy, inicialmente com zeros, tanto faz, de tamanho nboot. Mil no default aqui da função. Eu vou fazer mil vezes nesse for aqui. Tirar uma amostra com o np-random. Tirar uma amostra com o np-random. With a function Shallow fires from the bred-out array dos meus dados, de tamanho n. Com reposição, aí eu calculo a média dessa amostra. Calculo a média aqui, e jogo ela na posição correta do... do... do vetor medias. E, aí essa função vai retornar um vetor de medias. Pode rodar. Gente, rodando essa próxima função aqui. amostras, amostra Bootstrap, está aqui. A cada ponto desse há uma média da minha variável x. E aí tem uma variabilidade, né? Claro, a gente sorteou, são numéricamente diferentes, mas tem uma variabilidade. A ideia disso é que a minha amostra é uma aproximação da população, é a melhor ideia que eu tenho da população. Então, se eu sortear da minha amostra, eu tomo isso como um sucedâneo, o melhor substituto de amostrar de novo da população. Essa é a ideia do Bootstrap. Daí, a gente vai aqui embaixo fazer um histograma para dar uma olhada na distribuição e a estética do histograma, kit, título e tal. E embaixo calcular o desvio padrão. fazer uma mensageria. O nosso histograma, pode rodar essa última célula, o nosso bonito histograma está aqui e a nossa mensageria está aqui. O desvio padrão das médias bootstrap é 0,11368, etc. Esse é o famoso algoritmo bootstrapping. Vamos lá, e o pegging? E agora? Então a ideia é a gente pegar a base, retira uma amostra bootstrap, treina o modelo, que vai me dar uma previsão. Tira outra amostra, treina outro modelo, que vai me dar outra previsão. Tira N amostras, vou ter M modelos, e a previsão final vai ser a média de todas essas previsões anteriores. Então daí vem o nome bootstrapping, o nome pegging, é um bootstrap aggregation. Aí junta tudo pegging. O random forest é um pegging de árvores. Então a gente vai tirar uma amostra bootstrap, rodar uma árvore, não é qualquer modelo, é uma árvore. Aí roda outra árvore e guarda. M de árvores. Jiangxi guarda todas elas. O padrão do Python, do scikit-learn, se eu não me engano, ele faz acho que 200 árvores. É árvore pra chuchu. Um monte de árvores. Então esse é o random forest. Run Forest, Random Forest, Random, muito ruim, foi a minha. Vamos rodar o nosso, vamos para o algoritmo 2.3 Begging. Vamos lá. Ju, tem alguma pergunta, angústia, dúvidas? Ju, deixa eu ver aqui, sim, já vou ser. Olha, o pessoal está mais tranquilinho, mas a Mariana Silva, lá de Salvador da Bahia, ela perguntou o seguinte, ela está uns minutinhos atrasada na aula, mas se poderia responder para ela, a partir de que critérios definimos uma profundidade? De duas ramificações, quatro ramificações? Então, é uma boa pergunta, é a pergunta de milhões, mas que a gente resolve fazendo o grid search. Qual que é o melhor número de ramificações? É aquele que dá o melhor desempenho na base de teste. E como é que a gente acha? Fazendo o grid search, que é isso que a gente viu até aqui. Nesse código, eu deixei um install Petsy. O Petsy é um pacote. O Petsy é um pacote que, olha, até onde eu saiba, só eu conheço. Não, só eu não, eu aprendi no livro do Wes McKinney. Mas eu gostei muito, assim, porque ele ajuda a usar, para quem conhece alguma coisa de R, o R tem uma coisa super bonitinha, que é o jeito que a gente define um modelo no R, a variável target e as variáveis explicativas. Aí o Petsy é isso para o Python. Então, vamos carregar as bibliotecas. Deixa eu ver se tem alguma novidade. O Random Forest aqui, que a gente vai usar, ele fica no pacote Ensemble do Scikit. Tem outros Ensembles aí também. Mas o Random Forest, eu acho que é o mais famoso, ficou super famoso nas competições de Kaggle, e aí depois ele foi ultrapassado pelo XGBoosting. Mas o próprio cara que inventou ficou surpreso. Ele não achava que ia dar um resultado tão bom. O Grid Search, nossa funçãozinha aqui de avaliar classificador. O Petsy e o Time para a gente cronometrar os nossos algoritmos aqui. Então, vamos carregar a base Titanic, né? Titanic para a gente não gastar tempo neste momento entendendo uma base nova. Titanic já é a nossa velha conhecida. Olha o Petsy aqui. Eu vou usar uma função do Petsy chamada The Matrices. Ele transforma esta anotação... Isso aqui eu acho muito bonitinho. Essa anotação em uma... Deixa eu aumentar aqui. Ele transforma isso em uma matriz de design. The Matrices é de design. Ele faz as matrizes de design. O que são essas matrizes? Vamos... Deixa eu mostrar aqui. Primeiro vamos dar uma olhada na carinha. A matriz X é a matriz de dados, a matriz de features. A matriz X que a gente está acostumado, né? Os estatísticos chamam isso de matriz de design, design matrix. E o Y também, né? O Y é a minha variável 3. O Y é a minha variável target ou variável resposta. Vamos dar uma olhadinha aqui no nosso explorador de variáveis. O X... Eu chamei de X grande aqui. É a matriz de design. É igualzinho. É igual o DataFrame mesmo. Só que tem essas variáveis que eu colocar aqui. Outra coisa legal. Está vendo o sexo aqui? Vamos ver como é que ficou o sexo. É... Aqui, ó. Sex T Male. É... Ele... Ele mudou um pouquinho o nome dela. Sex T Male. T é de treatment. É uma anotação estatística. Mas o male quer dizer que o 1 é o masculino e o 0 é o feminino. E aí assim por diante, né? Embarked. Treatment. Q. Q ela está indicando que é Queensland. Então os indivíduos com 1 são de Queensland. Então é um jeito super legal de criar os dummies para a gente. É um jeito bem prático e... Não sei... Gostosinho, né? Não erre. Isso aqui faz um sucesso danado. A turma adora. Assim... No Python, de fato, a gente tem um passinho aqui a mais. Faz a matriz. Depois usa o método. Não erre. A gente... O método já com a anotação ali. Já fica... Mais... Mais... Um pouquinho mais sintético. A gente indica a base de dados. E o return type para ele retornar um data frame. Porque senão ele retorna um objeto do numpy. E aí a gente pode se atrapalhar um pouquinho. Né? Então... Legal. O d matrices. E aí a gente roda... Divide o train test split. Né? E... Já fizemos o... A nossa divisão como tradicional. Né? Dividimos 596 indivíduos para... É... É... É... Treino. 143 para teste. E... A... É... Nove variáveis que a gente vai usar. Vamos fazer o nosso primeiro random forest. Vamos fazer 50 árvores para classificar o titanic. O titanic é bom porque ele... O titanic é bom porque ele é pequenininho. Não judia do hardware de vocês aí. E... Aí a gente pode por um random state aí. Para a gente ter mais ou menos a mesma coisa. Pode rodar. É... Você vê que ele roda rápido. Né? 50 árvores. Olha o meu aqui. Tá bom que o meu PC tem um processador bastante atualizado. Mas roda rápido mesmo. É... Aí no de baixo aqui. A gente vai avaliar a árvore. Na base de treino. E na base de teste. É... Vamos ver as figurinhas. Ó... A árvore... É... Solta né? Um... Uma mensageria aqui. A curaça da árvore. É... Curaça balanceada. AUC. Gine. Um gine excelente na base de treino. Gine na base de teste caiu... Bastantinho né? Mas... Não tá... Ruim. É... Nossa base na... É... É... É... É... É... Nossa base na... É... É... A matriz de confusão na base de treino. Muito bom pra ser verdade. A curva rock na base de treino. É... Muito bom ainda pra ser verdade. E... É... Quando a gente olha na base de teste já é uma coisa mais realista. Essa já é uma carinha mais normal de curva rock. Aí fica a pergunta. Será que a gente não consegue melhor? Mas... Olha só. Com muito pouco esforço a gente já conseguiu 75 de gine. Já tá assim... Tudo no default praticamente né? É... Então vamos fazer um grid search aqui. Com o random forest agora. A gente vai por um cronômetro aqui né? É... Tempo início, tempo fim. Ele vai... É... Imprimir uma mensageria aqui. É... Tempo início, tempo fim. Ele vai... É... Imprimir uma mensageria aqui. É... Tempo início, tempo fim. É... Tempo início, tempo fim. É... Tempo início, tempo fim. Ele vai... Tempo início, tempo fim. Ele vai... Com o tempo de execução do algoritmo. É... Vamos lá. Resgatar todos os elementos aqui. ParamGrid. Aqui eu vou indicar o parâmetro que eu quero buscar, né? Otimizar. Por exemplo, número de estimadores. Número de estimadores ou número de árvores? Que a gente vai fazer. Número de iterações no bootstrap. Eu vou por 100. 100, assim eu poderia colocar, sei lá, 100 e 200 aqui pra fazer mais ali eu ia fazer todas as combinações com 100 max features ele ia usar de 1 até 10 lembra, o range não entra no máximo, isso aqui é de 1 até 10 ele ia fazer 100 com max features igual a 1 100 com max features igual a 2 até 10, depois 200 com max features igual a 1 depois 2, depois até 10 por isso que se chama grid search, porque ele cruza todas as possibilidades que você colocar aqui então cuidado com o que você coloca aqui, eu vou deixar aqui igual o tava original pra gente poder rodar igual aí eu vou definir um objeto que eu estou chamando de grid search, que ele é da classe do grid search eu vou definir esse parâmetro aqui, estimator é o random forest que eu defini lá em cima o parangrid é o grid de parâmetros que a gente definiu logo aqui o scoring é o que ele vai usar pra escolher o melhor modelo, o ROC é o C vou fazer um cross validation de 4 dobras e usar todos os cores do meu processador se ele tiver vários cores tá aí, a gente roda o fit aí eu vou querer ver o que que tem nesse objeto, os melhores parâmetros o melhor score, o melhor score é o melhor ao C, o melhor é esse aqui se eu mudar pra curácea isso aqui vira curácea e aí ele vai imprimir o pode rodar isso aqui, ele vai imprimir o o tempo de execução do algoritmo ó, no meu aqui deu 2 segundos e 85 traz aí pra gente quanto tempo demorou de vocês, que eu fico curioso, as vezes tem gente com um computadorzinho bem simplesinho e demora um pouquinho mais coloca aí pra mim quanto tempo demorou de vocês o próximo a próxima etapa aqui a gente vai rodar o avalia CLF na base de treino e na base de teste com o melhor modelo, lembra o que é o melhor modelo? é o Best Estimator esse aqui já é ele já segue a cartilinha do Tibichirane ele já reestimou o melhor modelo na base de treino inteira então a gente pode usar esse carinha aqui o Grid Search já faz aquilo pra gente vamos ver como é que ficou o nosso modelo aqui, nosso Best modelo então a AUC na base de treino ficou bem ignorante, não sabe brincar e 99 e 97 de AUC o Gini bem alto 77 e 89 de Gini na base de teste bom, a gente pode colocar mais coisas no Grid, ah é eu não falei o que é Max Features essa é uma boa pergunta que vocês devem muitos de vocês devem estar se perguntando assim como o deixa eu voltar aqui nesse desenho assim como o Bootstrap sorteia as linhas Max Features sorteia as colunas então Max Features igual a 1 ele vai fazer 100 árvores com 1 Feature na árvore número de Features igual a 2 ele vai fazer 100 árvores com 2 Features que ele vai sortear N Features igual a 3 com 3 Features e assim sucessivamente vamos ver qual que foi esse é um elemento importante do Random Forest isso faz ele ficar eficiente e e melhora ele curiosamente a gente tirar algumas variáveis ajuda a criar diversidade é um parâmetro que dá para melhorar a árvore vamos ver qual que foi o eu não olhei aqui em cima tinha olhado só o tempo de execução aqui vou voltar um pouquinho então o Max Features ótimo foi 5 não foi o máximo foi 5 bom o número de estimadores era 100 que eu só rodei com 100 mesmo bom o que tem no GridSearchCV ele mostra os parâmetros aqui que a gente rodou o Random Forest ele coloca todos os parâmetros ali bonitinho ok então esse é o Random Forest e tunado com GridSearchCV o Random Forest é baseado em árvore então ele tem algumas novidades como o número de estimadores e Max Features que não vai ter no algoritmo clássico de árvore mas ele tem o CCP Alpha também ele tem a profundidade máxima dá para a gente colocar tudo isso definir árvores mais simples mais complexas enfim fazer o nosso GridSearchCV ok então esse é o nosso Random Forest tem dois algoritmos no livro que eu vi acho que era um Tibichirani mesmo ele define esse nome Pesting em geral a galera não usa esse nome mas o Begging tem algumas sutis diferenças mas no algoritmo o Begging faz o sorteio com reposição o Pesting faz sem reposição e aí o tamanho é um Q menor do que N tem que ser né o Begging é tamanho N o Scikit Learn deixa tem esses parâmetros aqui também se a gente procurar um pouquinho e no chat repetir a gente descobre fácil como faz como um desses parâmetros ele constrói um modelo é bom é igual né na amostra e repete a diferença é que um é com reposição e o outro é sem reposição é levantar características ele roda em paralelo já que ele vai fazer 100 árvores se eu tiver dois processadores eu boto cada um para trabalhar uma árvore aí termina cada um vai trabalhar outra árvore eu vou ganhar o dobro vou fazer na metade do tempo se eu tiver sei lá aqueles 32 cores dá para distribuir o processamento geral ali né ele também classifica em paralelo já que eu preciso classificar com 100 árvores ele também pode deixar cada core treinando uma árvore também ganha eficiência e costuma ter bom desempenho sem grandes ajustes ele é razoavelmente robusto a um monte de coisas a outliers nas features ele é robusto a é , dados de tipos diferentes ele vai bem com dado contínuo dado discreto dado qualitativo ele faz regressão faz classificação faz um algoritmo por isso que eu gosto de comparar porque eu gosto de dizer que ele é um GMC Hummer aquele carrão um negócio robusto poderoso que vai em qualquer terreno que corre vai bem na lama vai bem no asfalto e tudo né as dúvidas que eu tinha quando eu aprendi o random forest se eu fizer 500 árvores o default do programa que eu usava era 500 o do scikit acho que é 200 assim ele vai fazer 500 árvores mesmo não é muito é ele vai fazer 500 árvores ele vai guardar as 500 árvores é isso mesmo é isso mesmo ele vai guardar as 500 árvores ele demora para treinar bom a gente teve uma boa percepção aqui né aliás Ju vê se alguém retornou aí com o tempo que demorou o algoritmo da turma sim pro muitos retornaram e ó vou falar para você recebemos uma enxurrada aqui de mensagens o Jorge lá de Fernando Prestes 1.54 a Yasmin no livro em 19 segundos rodou da Yasmin mais que eu aí no caso esse Jorge deve ter aquele pc gamer o cara deve ter aquele pc gamer top em que você joga aí GTA 5 né que mais desculpa o Gustavo levou 5 segundos e 77 centésimos o Gustavo levou 5 segundos e 77 centésimos o André 9 segundos é mais parecido tá dentro aí é legal 0.7 segundos isso eu não vou jogar counter strike contra você não vamos lá bom assim demora para treinar demora isso aí numa base maior claro que vai demorar mais mas a turma roda lá no banco sem problema e as vezes a gente usa a WS razoável assim as vezes pior que o meu notebook roda é surpreendente o tanto de coisa que a gente consegue rodar aí não roda tá na nuvem a gente pede mais memória aí ele anda melhor bom e para aplicar a regra a gente tem que aplicar tudo isso de regra 500 regras demora assim dependendo da aplicação pode até chegar a impactar tem aplicação que a resposta tem que vir em real time ou em near real time isso quer dizer que a resposta tem que vir muito muito muito rápido o usuário não pode perceber muito a demora senão ele vai ficar irritado é assim pode começar a impactar mas dependendo do equipamento dá para colocar e aí o algoritmo guarda tudo isso de árvore ele guarda sim pode parecer bastante mas em geral ele vai bem bom é isso é um GMC Hummer roda em paralelo classifica em paralelo tem bons desempenhos robusto outliers na variável resposta não é tão verdade mas ele acaba tendo uma certa robustez também como aleatoriamente o outlier entra e o outlier sai e você está avaliando na base de teste mesmo ele é influenciado pelo outlier mas não é uma coisa em geral tão crítica o outlier também em geral vai ficar numa folhinha lá dele e aí o impacto é menor do que em outras técnicas que a gente usa bom vamos dar uma olhada aqui bom as sugestões também para a gente calma aí deixa eu ver se a gente vamos voltar pro nosso é é spider aqui então a gente rodou esse bagging eu vou deixar aqui também uma é deixa aqui um exercício pra gente dar uma olhada que ele é um exercício que eu acho bem legal esse aqui acho que é interessante da gente explorar que ele é é é um exercício que eu gosto bastante é a gente é como se a gente tem assim um um celular no bolso a pessoa está com o o indivíduo lá do teste está com o celular no bolso ele até fala qual é um samsung galaxy velho lá que é antigo estudo e é aí ele está tentando adivinhar o que que o indivíduo está fazendo ele pode estar subindo escada descendo escada pode estar fazendo um monte de coisas aí né deixa eu só mostrar aqui a título de curiosidade onde está essa base é muito fácil de achar se a gente colocar é human activity recognition é é é , é uma base do UCI aí o primeiro link aqui human activity recognition esse portal tem várias bases interessantes aí para procurar essa daqui é uma você pode baixar as bases daqui são bases para machine learning da universidade irvine é essa base é uma das famosonas aqui a gente pode baixar os dados ele explica o que são os dados de tudo e essa é uma basezinha é bem legal eu deixei aqui um código que carrega essa base é e deixei também é é bom a gente já vai é passar ele né eu deixei também a base é pronta para se a gente precisar né então vamos lá a gente vai vou ler a base aqui a base deve estar junto opa eu esqueci de rodar aqui a gente vai usar o pandas e o OS aqui no primeiro momento então é eu como é que vem essa base as vezes a gente tem um probleminha maior de tratamento de dados né ela vem com as features num lugar é a base de treino no outro a base de teste no outro a target no outro e enfim aqui a gente carregou as features aí tem um tratamentinho de dados aqui que a gente vai precisar fazer né então eu estou as features aliás elas vem com parênteses vem com né caracteres não imprimíveis tal vírgula e tudo isso vai atrapalhar a gente horrores né então eu deixei um tratamentinho aqui para limpar o nome das features pode rodar esse cara aqui e aí vamos vamos ver como é que estão os nomes das features aqui chama features path é um objetinho tranquilinho né ele é o caminho só features ele não está abrindo aqui ai não é esse features ele é uma lista ele é um objeto tipo lista ele tem o nome das features gente a gente tem quinhentas e sessenta features sem exagero é como é que tem tanta features o acelerômetro e o giroscópio do celular do cidadão vão gravando dados a cada quarenta hertz se eu não me engano com a frequência de quarenta hertz ou seja quarenta pontos por segundo ai eles pegam uma janela janelas né acho que é um segundo e alguma coisinha e vão tirando estatísticas né a média aceleração corporal média no eixo x aceleração corporal média no eixo y aceleração corporal média no eixo z tá como é que você vai arrumar criatividade para dar quinhentas calma ai desvia o padrão média eu não lembro o que que é ele tem uma definição lá que ele explica o que que é é o máximo aceleração máxima no eixo x naquele intervalinho no eixo x no y e no z aceleração corporal mínima no intervalo assim o mínimo por exemplo pode ser interessante porque se a pessoa tiver correndo já não vai ser muito baixo o acelerômetro e o giroscópio vão estar doidos ali é sma não lembro se é simple moving average ai ele tem umas medidas de energia né são as derivadas do acelerômetro do giroscópio ai tem o jerk que seria assim a gente tem física cinemática tem o deriva a posição deriva a posição no tempo a gente tem a velocidade deriva a velocidade tem aceleração deriva a aceleração tem esse tal de jerk que vai aparecer aqui no meio a entropia coeficientes aqui gravidade enfim tem um monte de indicadores ai eles vão fazendo combinações em cima de combinações da quinhentas features é esse o exemplinho que a gente vai mexer agora é bom rodamos até aqui só o nome das features aqui a gente tratou os nomes para eles ficarem mais bonitinhos tirar e eu estou colocando aqui vocês devem conhecer o f string o list comprehension deixei um list comprehension para a gente pegar cada posição no meu set de features e é e e colocar um índicezinho na frente eu vou colocar o i com 3 posições para poder caber os 500 tracinho o nome da feature e ai vai ficar assim ó nomes variáveis 000 body acc minx para eu saber no nome a posição e do que se trata a variável acho que já está meio que na hora do recreio mas deixa eu só terminar essa partezinha aqui é aqui a gente está verificando se tem duplicações né achamos 84 duplicações mas nesse caso aqui eu não tiraria as duplicações mesmo acho que deve ser fortuito ali pela precisão do do aparelho mas caberia a gente investigar se essa duplicação é uma duplicação mesmo ou se é fortuito eu acho que não é uma duplicação mesmo porque o tempo vai mudando né pode ser uma o samsung do cara deve ter dado uma travadinha ali e acho que está dentro bom ai a gente vai definir o objeto x o caminho do arquivo né ai aqui a gente vai pegar o caminho do a gente tem as variáveis vão estar nesse arquivo xtrain os indivíduos a gente tem acho que 30 indivíduos 50 indivíduos na amostra tem o número do indivíduo aqui se a gente for usar para alguma coisa eu estou carregando aqui ai o xtrain é eu vou ler o csv é e sem header né o nome das features ele vem separado ele é um meio chatinho esse arquivinho aqui mas eu deixei todo o tratamentinho aqui que tem uns recursos interessantes é ai o subject train a gente vai ler num outro arquivo e depois vai juntar os dois o xtrain a gente vai fazer e deixar o subject como índice da tabela ai o índice serve para alguma coisa nesse caso aqui eu vou usar o pdconcat e com axis igual a 1 ele concatena de ladinho assim e e vai ficar legal vamos rodar isso aqui ai já vou abrir objeto por objeto para a gente entender xtrain como é que ele é ele já está com o subject aqui já está concatenado essa variável não vem eu que concatenei é ai cada uma das variáveis ai tem um monte de variáveis uma base grandinha e ela tem um monte de linha também ela é uma base razoável de grande mas em geral tratável é e ai aqui a gente vai ler a base de é os y a mesma coisa a gente vai ler os y é os y vamos rodar essa célula aqui a gente tem os labels num lugar os labels e o y mesmo então a gente lê os labels labels diz que é andando dois é subindo três é descendo e assim por diante ai a gente vai definir os activity labels e marcar ler o objeto y e marcar os labels nele desse jeito a gente vai ter um dicionário chamado activity labels que é este ele está indicando um é walking dois é walking upstairs três walking downstairs sitting standing laying então tem três atividades mais ou menos estáticas três atividades mais ou menos dinâmicas Ju só terminar essa partezinha aqui que ainda está meio no meio esse activity labels a gente leu de um arquivo e o y mesmo leu de outro arquivo desse arquivo y train vamos ver como é que está o y aqui o y não é este é o y train este é o meu y ai ele já tem uma variável tipo category do pandas porque a gente definiu assim aqui ó aqui o y ele vai ser um category que é o próprio y originalmente ele é um numerinho mas eu vou dar um label para ele que é o de acordo com activity labels e ai a variável tipo category ela ocupa pouquíssimo espaço e guarda um rótulo grande mas que se repete toda hora então a gente vai ter esse indivíduo aqui estava de pé durante toda essa parte do experimento e ai ele sentou e ai ele deitou e ai ele estava andando e ai ele vai recebendo instruções aleatórias para ir fazendo atividades e ai o nosso objetivo é botar o algoritmo para adivinhar o que ele está fazendo então eu acho que é um tipo de de problema que ele foge um pouco de classificações super tradicionais de marketing, de crédito ela é uma coisa mais a ver até com como é que eu desenho um wearable que ajuda o cara no treino dele de triatlon diz Ju, você ia falar alguma coisa? acho que dá para ir para o recreio agora não pro falei que tudo bem, que o senhor podia concluir depois nós iríamos, fique a vontade não, mas acho que até aqui já está bom a gente retoma aqui o resto desse código e avança mais um pouquinho vamos para o recreio? então vamos lá pessoal vamos ao intervalo vamos tomar um café, uma água e a gente retorna daqui 15 minutos com a rodada de perguntas que o professor Juca vai responder, está bem? então até já já

BLOCO 3

Pessoal, vamos voltando para o nosso último bloco da noite. E antes de iniciar a rodada de perguntas, eu vou passar a palavra para o professor Juca. Oi, vamos lá. Deixa eu começar aqui, porque eu vi que teve uma pergunta sobre o link da animação da Valéria Miranda. Bom, ela está pública aqui, né? Eu já me certifiquei de que ela está pública. Se alguém, por acaso, não conseguir... Eu mandei no chat essa pergunta. A pergunta era essa. Eu acho que você pode mandar de novo no chat, Ju. Mas eu vou passar aqui que vale a pena, né? É, bom, um memezinho para não usar as amostras de teste para avaliar a qualidade do modelo. Você usa a validação cruzada para isso. A ideia do K-Fold, né? A gente divide nos grupinhos, em quatro grupinhos. Primeira etapa. A gente divide em K grupos. No exemplo, K igual a 4. Segunda etapa. A gente define uma participação, por exemplo, essa. E treina o modelo com as demais. Aí a gente classifica a amostra de validação com esse modelo que a gente treinou nessa basezinha aqui. A gente volta ele para cá. Treina o modelo aqui. E aí aplica ele, faz a previsão dessas observações aqui. Aí a quinta etapa. Ô, o senhor não está compartilhando sua tela. Jesus. Vamos lá. Desculpa. É uma animação tão bonitinha. Eu crente que estava passando. Aqui é o memezinho. Vamos lá. Volta a fita. O memezinho aqui usa a amostra de validação cruzada para isso. Tá? Não. É bom. Então, a primeira etapa do modelo é dividir a base de dados em quatro grupos, K grupos, no caso 4. Aí a gente vai definir uma partição para usar de validação, as outras para treinar o modelo. Aí a gente vai aplicar esse modelo nessa base e vai ter uma avaliação dele. A gente vai repetir os passos 1 a 5, só que cada hora separando uma base diferente de validação. E vai fazer exatamente os mesmos passos. Daí a gente vai aplicar o modelo, avaliar o modelo 1 em cada uma das bases, calcular a média da medida de avaliação, por exemplo, a curácia. E aí a gente vai fazer o mesmo para todas as bases. E aí a gente vai fazer o mesmo para todas as bases. E aí a gente vai fazer o mesmo para todas as bases. E aí a gente vai fazer o mesmo para todas as bases. Vamos avaliar na base 1, na base 2, na base 3, na base 4. Tem uma curácia média, no caso... É isso. Uma ilustração bonitinha do Cafoldi. Aí vocês guardem ela e podem consultar depois se quiser. E é isso. Vamos para as perguntas, Jô? Vamos sim. Vamos começar aqui a rodada de perguntas. Tem umas perguntas. que eu achei muito boas. Agradeço as perguntas. Imagina. Pronto. O Fabiano, lá de Brasília, fez a seguinte pergunta. Professor, como podemos obter a explicabilidade do modelo de Randolph-Forrest, tendo em vista que, a depender do experimento, será necessário explicar quais atributos influenciaram na probabilidade de ocorrência do evento. Por exemplo, na avaliação de risco de crédito, o que levou o cliente a ser reprovado, ficar abaixo do ponto de corte para a contratação? Essa é uma boa pergunta, até porque cada vez mais a gente tem leis que protegem o consumidor. E aí, se alguém usa alguma informação nossa para alguma coisa, a empresa tem que conseguir responder por aquilo. E no crédito, no caso, dentre numerosos, dentre as outras aplicações, a empresa tem que conseguir falar motivos em muitos lugares, muitas vezes. A árvore já é difícil de interpretar quando ela começa a ficar muito profunda. O Randolph-Forrest piorou, porque vai ser um monte de árvores, não vai... vai ficar praticamente... vai ficar completamente impossível de explicar cada uma das árvores. Aí tem estratégias de fazer isso. Estratégias desde as mais básicas, que é como fazer uma análise descritiva. Ver, por exemplo, a média da predição do modelo para cada nível de uma variável que você tenha na base. Aí... Por exemplo, por tempo de emprego, né? A média de previsão do modelo por tempo de emprego. Aí, se tiver uma relação interessante, vai aparecer ali. O modelo está indicando... É o que o modelo está falando, não o que é de verdade. O modelo está falando que quanto maior o tempo de emprego, menor a taxa de inadimplência, por exemplo, ou... Sei lá, o problema que você estiver analisando. É... A... Essa é uma ideia bem básica e tem ideias baseadas nela usando uma técnica de regressão, que também não está nem aqui no nosso escopo, né? Assim, a gente faz um modelo mais simples e mais interpretável que ajudaria a gente a interpretar um modelo de machine learning. E tem a técnica mais... É... Não sei se popular, talvez, hoje em dia, que é do Nobel de Economia, de, acho que, 2017. 2017 foi o Richard Thaler. Esse foi antes. Que é o Shapley. Ele tem uma técnica que o pessoal chama de Shap. É... Bom, eu vou... dar uma olhada aqui, que é fácil da gente achar uma referência na internet sobre o Shapley. É... Shap... É... Machine Learning... Explanation... Shap. Uma pesquisa assim já deve responder... Ó, tudo isso são coisas que o Shap indica, né? A gente tem... É... Deixa eu pegar se tem um... DataCamp... Eu gosto do Medium... Bom, mas tem vários legais aqui. Vou pegar esse Geeks4Geek, que, em geral, ele tem uns artigos legais também. Ele explica que... Vou deixar esse... link... Deixa eu pegar o gráfico que ele faz, ó. É... Ele pega... Aqui... No nível indivíduo. Para um indivíduo, ele está falando... Olha, para este indivíduo, essa variável, se eu tirasse ela do modelo, ia ter um impacto... É... É... É... Esse impacto aqui é atribuível a essa variável. Como é que ele faz isso? Ele dá a previsão do modelo com a variável, e sem a variável, ele pega a diferença. Então, esse é o... Ele diz o efeito atribuível a esta variável, que deste indivíduo, no caso, tem um valor específico. Né? Por exemplo, é... Que é... ShellWeight. Ele está explicando aí o... O exemplo direitinho. De acordo com o peso da concha, ele vai indicar a resposta aqui do modelo, que... Eu não sei o que que é, mas... Por causa do peso da concha, a variável resposta teve esse efeito. Por causa do... Peso total, o efeito foi esse outro aqui. E aí, ele vai compondo todos os efeitos. Ele vai indicando que efeito que deu cada uma das variáveis. É... Positivo, ele está deixando esse rosinha aqui. E negativo, ele está deixando esse azulzinho aqui. É... Ele é interessante. Ele dá para fazer análise no nível indivíduo, e tem esse tipo de análise no nível população. Né? Que aí eles colocam os efeitos de... Cada pontinho aqui é um indivíduo, é... Que teve um efeito para um lado ou para o outro, de acordo com... Com a variável, né? Então, a gente consegue tirar explicações no nível microscópico do indivíduo, e no nível macro da... É... Da população também. Dá para ver as variáveis mais importantes. A tela. Ai, não. De novo, eu fiz aquilo. De novo. Nossa. Por hoje, vou compartilhar a tela. Posso só dar um recadinho para os alunos? Pode. Pessoal, vocês estão enviando mensagens sobre o link de permissão de acesso ao conteúdo do link que o professor enviou. É só vocês acessarem com a conta Google de vocês, certo? Cadastrarem o acesso com a data de aniversário, a cadastrar com a data de aniversário de vocês, com o estudante que vocês têm o acesso. Certinho? Quem tem aí, volte para a gente. Prô, a palavra é sua. Ó, deixei o link aqui do artiguinho. Ó, no nível indivíduo, ele dá até uns gráficos bonitinhos, né? É... Tem essa variável shell weight, ela dá um efeito... É... Assim, né? O efeito em geral deve depender do valor da variável. Então, ele indica o valor da variável e o efeito. Aí, num problema de risco de crédito, por exemplo, ele diria alguma coisa como... Essa variável tem esse impacto na probabilidade de inadimplência. Essa outra tem esse outro impacto. Essa outra tem esse outro. E aí, por aí vai. Ele vai compondo os efeitos de cada variável. No nível população, ele tem um gráfico assim, né? De como é que é o efeito, o impacto no output do modelo, né? De cada indivíduo aqui para cada variável. Então, quanto mais espalhado, em geral, mais importante a variável é. Né? Então, é um... Isso aqui que é essa... A técnica mais utilizada de... É... De procurar explicar modelos de machine learning. É o CHAP. Acho que responde bem a esse tipo de pergunta mesmo. Aí, dá uma olhadinha no artigo. Tem todos os passos aí para usar ele, para aplicar o CHAP. Ele é bem legal. O que mais? Tem mais perguntas boas aí. Vamos lá, professor. Vamos para a próxima pergunta. É... Agora, para a pergunta do Benjamin. Para o RendoForest, não precisa buscar o CCP Underline Max para obter o melhor modelo? É... Sim. Pode. Né? É um parâmetro a se tunar da... O CCP Max... CCP... CCP Max. Não, calma. Deixa eu voltar no código aqui, para a gente... Só garantir que a gente está falando da mesma coisa. É... Quando a gente montou o nosso Grid Search, é... A gente... O... O outro, da regressão, né? A gente... É... Definiu o Grid com base no CCP Alpha. O nome do parâmetro é CCP Alpha. É... Aí, a gente definiu o Grid. É... Treinou o Grid. É... E pegou o... É... E pegou os melhores parâmetros aqui nessa linha. É... O CCP Max, acho que foi um parâmetro que eu defini. É... Foi no... No outro código, né? Eu coloquei... É... Não lembro onde estava aqui. Eu... Eu criei uma variável que continha o CCP Max. Né? É... E aí, o CCP Max é o máximo do CCP Alpha que o Grid Search vai achar para gente. Aí, por exemplo, nessa linha aqui, ele indica a melhor combinação de hiperparâmetros. Só tem o CCP Alpha. Ele fala o CCP Alpha que nos dá o... A melhor... É... Árvore. Né? É... A melhor... Melhor árvore mesmo. É... É isso. É... Foi no... No... No outro... No outro caso aqui. Na outra... Aqui, ó. CCP Max. É o CP... É o... Eu criei uma... Aqui tem que dar uma olhadinha no código com um pouquinho mais de calma. Ó, um data frame. A gente fez o Grid Search na mão aqui, né? A gente não precisa fazer na mão. A gente pode usar... É... Eu gosto de fazer na mão para a gente controlar todos os elementos do algoritmo. Uma vez, pelo menos. Aí, aprendendo a fazer na mão, a gente pode usar o Grid Search CV. O Grid Search CV já faz... Já... Cobre o escanteio e corre para cabecear. Já faz tudo. Ele... Define... A gente define o Grid. Define o Grid Search CV e quais vão... Quais vão ser as características dele. É... Qual é o modelo que eu quero tunar? É esse. Qual que é o... Quais são os parâmetros? Esses. Qual é o método de validação cruzada? É esse aqui. No caso, eu estou pedindo um K-fold CV5. É Cross Validation com 5 folds, né? Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... Ele tem... ele guarda aí no objeto todos os resultados. Bom, daí, no Random Forest, a gente tem o CCP também. A gente tem esse mesmo parâmetro aqui, ó. Deixa eu ver onde é que ele tá, Bootstrapping, aqui, CCP Alpha. Esse parâmetro, a gente pode usar, a gente pode colocar no grid do Random Forest também, a complexidade da árvore. Aqui, a gente não mexeu na complexidade da árvore, só mexemos no máximo número de features da nossa árvore. É isto. Tem mais alguma, Ju? Prô, antes da gente partir para a próxima pergunta, os alunos estão se queixando do acesso limpo. Se eles não estão tendo permissão, se não está aparecendo para eles editarem. Eu vou dar uma olhada aqui, mas eu achei que, bom, para mim ele apareceu como público. Honestamente, eu não testei, mas não achei que era para dar problema. Esse aqui, aqui, ó, isso aqui, privacidade pública. Eu pude verificar, pessoal, professor. Aqui existe templates free, certo? Ah, esse aqui é público, né? Eu deveria, ele deveria estar público, né? Bom, compartilhar. Vou copiar esses dois links aqui que ele dá para mim. Vou colocar no chat. Vamos ver. Certo, vou compartilhar com o Zé. Esse outro link aqui. Se você puder compartilhar. Acho que... Vai dar para ver. Bom, aí. Assim, de qualquer forma, também tem o mesmo material nos slides em forma de PowerPoint. Uma coisa bem parecida. O que mais? Vamos lá para a próxima pergunta, então. Calma aí. A última pergunta, professor. A última não, temos duas. A Edilaine mencionou o seguinte, que ela não entendeu... O que são max-features. É, o max-features. É esse parâmetro do random forest. Vamos lá, deixa eu pegar um data frame aqui. Não, esse aqui está ruim. Deixa eu ver. Tem um train aqui. Algum pouquinho melhor. Por exemplo, o nosso X-Test. Pode ser esse aqui, só para exemplificar. O Bootstrap, ele vai sortear as linhas, normalmente. É o básico que ele faz. Além de sortear as linhas, ele também sorteia as variáveis. Então, se eu mandar fazer árvores com três variáveis, ele sorteia três dessas. Quais queres? Então, ele pode pegar... Se tiver um interceptor, ele pode pegar o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test, o X-Test. Se tiver um interceptor aqui, ele vai pegar um interceptor. Interceptor é um para todo mundo. É uma coisa que usa para fazer regressão. O Sex, Male, Embarked e Age. Ele vai fazer a árvore com essas três. Na próxima interação, ele sorteia de novo as linhas. Aí, ele sorteia, sei lá, P-Class, Part, Fair. Aí, ele sorteia de novo. Então, é um parâmetro que indica por quê? Para ele não usar todas as variáveis. Às vezes, ele usa uns subconjuntos de variáveis. E a variável mesmo, talvez uma determinada variável, Part, traga algum conhecimento dado o nível de alguma outra coisa. Então, aí o Random Forest vai agregando mais diversidade ao algoritmo. O que mais, Jo? Vamos lá para a próxima. O Fernando Lima perguntou. Minha pergunta não ficou clara. Quando pensamos em um modelo de regressão, geralmente, temos que ser mais parcimoniosos no número de variáveis. No caso de um Random Forest, essa preocupação não existe, certo? Essa é uma boa pergunta. O Feature Selection, tradicionalmente, na seleção de quais Features a gente vai usar no modelo. Tradicionalmente, é uma coisa importante no Machine Learning. Mas, tanto o Random Forest quanto o XGBoosting ou o LightGBM, que é o que está mais na última moda, que a galera mais usa. O LightGBM é um Gradient Boosting mais rápido. Eles são robustos para... Eles são robustos para ter muitas variáveis. Assim, a gente até discute bastante isso no meio profissional. Quando eu trabalhava com validação... Aliás, eu trabalho de novo com validação no Itaú. A gente pegava modelos que o cara coloca um monte de variável inútil. Aí a gente tira, faz de novo o modelo e dá o mesmo resultado. O resultado equivalente. Um empate técnico na qualidade dos dois modelos. A pergunta é, por que o cara botou tanta variável inútil? A resposta é, porque ele pode. É porque não vai fazer mal no desempenho do modelo. Talvez deixe um pouquinho mais lento para treinar. E talvez deixe um pouquinho mais lento para aplicar o modelo. Para fazer a previsão. Mas, assim... Em tese ele é bem robusto para Feature Selection. Ele roda bem com base grande. Vai super bem. No Lite GBM ainda, dá para não se preocupar com isso. Aí a pergunta é, por que você vai deixar tanta coisa? Não dá para fazer uma coisa mais enxuta? Que gasta menos memória, talvez? É nessa linha que vai a conversa. Não é na linha do desempenho. O desempenho, ele come com farofa as variáveis inúteis ali. Mais alguma? Não, foram essas. Tá. Bom, vamos lá então. Vamos seguir aqui, só para eu fechar esse algoritmo aqui. Esse código. Dessa base do UCI. A gente rodando esse código. O que a gente vai ter no final? X teste, X treino, X teste, Y treino, Y teste. X treino e X teste vão ter as 560 features. Acho que tem umas 10 mil linhas. Ele fala aqui 7352. E o teste tem 2000 e pouco. A soma dá 10 mil. E o Y tem, não é este. Y treino. Está indicando se o indivíduo está em pé, está deitado. A atividade que ele está fazendo. Daí, essa base aqui. Bom, eu vou rodar todas essas células. A gente rodando todas elas. A gente vai ter as bases prontinhas. De X treino, X teste, Y treino, Y teste. E aí, eu estou guardando em pickle. Uma diquinha aqui. Eu canso de receber base em CSV. Acho que não tem motivo para a pessoa me mandar um CSV. Ela pode salvar em parquet, pickle. Então, uma diquinha interessante aqui. Ele é mais rápido. O arquivo fica menor. Ele guarda todas as características do Python. O CSV tem que ficar se preocupando se é ponto e vírgula, se é vírgula. Se é o que for. Espera aí. Deixa eu pedir um segundinho de licença aqui. Oi. Minha filhinha aqui. Ela já está acabando. Eu já vou dar atenção para ela. Então, eu estou salvando em pickle. Para a gente poder continuar o trabalho. Aí, eu quero que vocês trabalhem para a aula que vem. Em cima dessa base aqui. Tentar desenvolver um algoritmo aqui. Para classificar ele aqui. Aí, vai ter um pouquinho de problema. Possivelmente, porque a base é um pouquinho maior. E aí, tragam na aula que vem. Que tipo de... É... De... Qual é que foi a experiência. A gente compartilha aí. Como é que seguiu o exercício de vocês. Tem mais alguma pergunta, Ju? Acho que tinha mais uma. Ou duas. Temos sim, professor. Temos uma pergunta do Wellington. O Wellington mandou assim para nós. Lá de Santos. Professor. Como o Boosting tenta otimizar os erros? Isso pode fazer dele... Um modelo com maior probabilidade para gerar o Overfitting? É... Aparentemente, sim. Mas o Overfitting depende do Tuning que a gente faz do modelo. Se a gente... Deixar o modelo ser feliz. Qualquer um vai dar Overfitting. Qualquer um. Qualquer um. É... Árvore. Regressão. É... Support Vector Machine. Naive Base. Qualquer um. É... A gente... Sabendo fazer o Tuning direitinho. É... A gente vai achar os hiperparâmetros. É... Certinhos. Que tunam aquele modelo. E aí... O XGBoosting. Nas competições de Kaggle. Quando ele saiu. O Random Forest. Quando saiu começou a ganhar um monte de competição. O XGBoosting saiu em cima. E começou a ganhar mais ainda. O XGBoosting começou a dominar. Dava sempre o Random Forest. O XGBoosting. O XGBoosting+. E aí ele ficou super famoso. E o... Aí depois saiu o Light GBM. Que é um XGBoosting. Mais rápido. E que gasta menos memória. Ou seja. Dá pra tunar melhor. Dá pra tunar mais rápido. Portanto, melhor. É... É... É isso. Assim. Ele tá sujeito ao overfitting. Mas pra isso a gente faz o tuning. É exatamente pra isso. E... Após o tuning. O... Light GBM. Acho que ainda é o... Queridinho. Assim. Da... Da... Da... Dos... Cientistas de dados. Até hoje. Tem mais? Professor. Tem sido Anselmo. Então. Para o Random Forest. O Light GBM. Não tem mais. Não tem mais. Não tem mais. Não tem mais. Parece que tem. Perdão. Pelaande. Presente. Never data. Bem agora rapear. Na sua... Whale. Doce. Então, o... Por s salary. Para o? Por quê? É... É baixo ser... Não. Não. Não. Muito... Eu acho que ele tá sujeito ao overfitting também. É... Não. Para shades. Dá uma à marca, né? Até aí com o light... Tá. Repete. não. Em qualquer algoritmo baseado em árvore, também não. Então, assim, por isso que é um GMC Hammer, né? Pode ter multicolinearidade, pode ter... O que é multicolinearidade, né? Duas variáveis features correlacionadas entre si, e aí em regressão, né? Vai dar um confundimento e dar problema pra... Sobretudo pra interpretar os parâmetros. Mas aqui a gente não vai interpretar, assim, ele atropela todas essas coisinhas mesmo, né? A multicolinearidade, outlier, tudo isso. Então, tem problema. Tem mais alguma? Por hora, não, professor. Foram essas. Tá, legal. Então, vamos lá, assim. O... Sugestões aqui, né? Eu deixei o... Um linkzinho aqui do UCI. O UCI tem no caso, essa base de dados do Human Activity Recognition, né? Usando smartphones. E aí tem um outro também que é de doença cardíaca, pra poder sair da temática aqui. Mas eu vou deixar uma outra sugestão pra gente retomar na aula que vem. Pra gente fechar aqui o tema da aula de hoje, né? Bom, a árvore é só o começo. Tem infinitas formas de a gente combinar modelos e essa que a gente viu até aqui é a mais famosa. Esses modelos são difíceis de interpretar. Aí a gente tem estratégias que ajudam a gente a interpretar o nosso modelo. O... Em métodos estatísticos, a gente usa um tal de um stepwise pra fazer... É... Escolher variáveis do modelo, né? O cross-validation ele tem uma certa analogia. É... Até tem alguns modelos que o pessoal faz cross-validation pra fazer feature selection. É uma ideia que tem lá no livrinho do Tibichirani. Sim. No caso de... Random Forest, esses modelinhos aqui, é colocar a variável num piora o modelo, né? Então... É... Fica... Assim, né? Comparar o desempenho do modelo na base de teste, dependendo do conjunto de variáveis, fica um exercício um pouquinho diferente. A gente vai estar falando de eficiência, né? Na feature selection. E aí... Pratique, né? Pra gente... É... Praticar... Assim, né? Seguir com os exercícios. Bom, é... Aí antes de fechar, eu pedi pro Wilson deixar pra vocês um outro projetinho pra gente retomar ele na aula que vem. É... Deixa eu abrir ele aqui. Ele deve ter deixado aí pra vocês. Ah, sim. Ele só não mandou junto, porque eu pedi um pouquinho mais em cima da hora aí, né? Então ele deixou... Ele colocou num projetinho extra. Que tem só o código que eu... É... Que eu deixei aqui pra vocês. É... Então... A gente... Esse código aqui... O que a gente vai fazer aqui? Vamos... Deixa eu rodar aqui junto com vocês, né? A gente vai... Só pra gente... Fechando aqui. Vamos carregar as bibliotecas aqui. Ah, esse código aqui vocês vão achar com o nome de... Atividade 1 da aula 2 enunciado. Ah, eu tô com o código errado aqui. Este. Agora sim. É... Isso. Então a gente vai carregar as bibliotecas. Seborn, Matplotlib pra... Seborn pra quê? Pra gente carregar a base Tips. E... Tips é uma base de gorjetas. Que ela vai ter essas colunas. Valor total da conta. Valor da gorjeta. Sexo do... Da pessoa que tá pagando a conta. Se a pessoa que tá pagando é fumante ou não. O dia da semana. Se almoça ou janta. O tamanho da mesa. É... Aí a gente vai construir essa variável aqui. Que é o... Percentual da gorjeta. E... Vamos dar uma olhadinha nessa variável. Aqui. Ela tem um pouquinho de outlier, né? Então uma situação opcional. Tem um outlier aqui esquisito com um percentual de gorjeta que é maior que o próprio valor da conta. Provavelmente o cara tomou um café de um dólar, deu uma nota de cinco e mandou ficar com o troco. E... Aí a gente pode querer filtrar esse outlier aqui. Nosso objetivo é ajustar um modelo buscando... Prever o valor do percentual da gorjeta. E aí vocês podem usar tudo que a gente aprendeu até aqui. Eu deixei um exercício. Também, de novo, uma base pequenininha, simples. Pra... Mas que é de regressão, né? Aí tem uma pesquisinha pra fazer, né? Uma... É... Assim, né? A gente tem o Decision Tree Classical. Tem o Decision Tree Regressor. Tem o Random Forest Classifier. Tem o Random Forest Regressor. A gente não mexeu com o Random Forest Regressor. Pode aproveitar ele aqui pra... Nesse exercício, que esse tem uma variável resposta contínua. Então... Aí eu tô deixando dois exercícios. Um que tem uma variável resposta... Os dois tem variável resposta diferente do que a gente tá acostumado. Esse, ela é contínua. E o outro, que é um exercício um pouquinho mais desafiador. Uma base um pouquinho maior e tal. A variável resposta é qualitativa e tem seis classes. E não só duas, como a gente tá acostumado. Ok? É... Ju, tô chegando no final do meu conteúdo aqui. Tem pergunta pra gente aproveitar aqui? Não, não dá. Aí a gente faz um wrap-up aqui no final do tempo, se não tiver nenhuma pergunta. Bruno, as perguntas que chegaram, os nossos professores já responderam. Tá, legal. Tá, então eu vou fazer um apanhado aqui, nesses minutinhos finais aqui, de tudo que a gente viu até aqui. Deixa eu pegar meu projetinho original original aqui. Vamos lá. A gente abriu com a lição de casa, que avalia um modelo de crédito. A gente fez uma árvore de regressão, de classificação, que foi o tema da aula passada. Avaliamos e fizemos deixa eu ver se aqui a gente já fez um grid search. Treinamos um modelo e fizemos um grid search na mão aqui, pra gente entender bem a ideia do grid search. Um grid search simples, sem CAFOLD, mas na mão. Aí na sequência, a gente fez um problema com a árvore de regressão, que eu queria dar essa experiência pra vocês de ter uma referência de como é que a gente faz um machine learning quando a target é contínua. Daí a gente bom, exemplinho aqui bem artificial, né? Mas ilustra bem o que a gente vai precisar, né? Tem vários problemas com target contínua, de previsão de renda, de na medicina e tudo mais. Então, vira e mexe aparece um problema. Curiosamente, eu achei que esse tipo de problema era mais frequente. Mas, no mercado, eu tenho a impressão que é o oposto. Problema do tipo classificação é muito mais frequente. A gente tem bem mais problemas de dizer se é inadimplente ou não, se o paciente vai morrer ou não, se ele vai curar ou não, se eu vou vender ou não. Tem muito mais perguntas de sim e não na vida do que de quanto vai dar o índice glicêmico depois da dieta ou coisa do tipo. Bom, aí a gente fez a árvore de regressão, usamos o Decision Tree Regressor, né? E aí a gente fez também um grid search aqui no final. O tipo de um grid search, né? Passo a passo testando a profundidade da árvore. A gente viu o algoritmo do bootstrapping, um código curtinho que faz uma reamostragem pra gente estimar o erro padrão de um parâmetro aí, uma média, por exemplo, um desvio padrão. E o bagging, o bagging ele é o bootstrap aggregation, é uma mistura de bootstrappings, né? Então, o... a gente, que é o Random Forest, que é o próximo código que a gente viu. Aí, aqui a gente já usou o grid search CV, que ele organiza toda essa estrutura do grid search que a gente viu, né? Então, a gente o que que fez aqui? Fez uma base de treino que a gente usou como holdout, né? O holdout, o que que a gente faz? Usa pra avaliar, qual vai ser a expectativa de resultado do modelo lá no final, quando a gente botar no valendo. E, no treino, a gente vai fazer o cross validation usando o k-fold. Na aula que vem, a gente vai ver uma pequena variação do k-fold, né? Que a gente pode usar também. O scikit tem o nk-fold, tem outras pequenas variações aí do k-fold que podem ser interessantes também. A gente define, fez uma primeira árvore e avaliou e o grid search, né? Como é que a gente faz? Define um grid, a gente vai colocar todos os parâmetros que tem dentro da função do nosso modelo, e uma lista de todas as possibilidades. Aí depois, o máximo número de features, o outro parâmetro, e todas as possibilidades que a gente quer testar. No caso, o max features, né? Aí a gente está testando aqui de 1 até 11. Eu vou fazer 10 modelos. Se eu colocar duas opções aqui, vão ser 30 modelos. Se eu colocar mais um parâmetro com duas opções, vai multiplicar por 2 de novo. Então vai sempre multiplicando o número de modelos que a gente vai testar no grid. E isso aqui dá pra ficar bastante complexo até, né? Bem grandinho. Eu deixei na aula grids pequenos pra gente ter tempo hábil de rodar em tempo de aula. Mas aí em tempo de projeto dá pra gente aumentar um pouquinho mais o grid search. Teve alguém que mencionou um método do random search. O random search, acho que vale mencionar também. Ele é interessante porque se a gente tem um grid muito, muito, muito amplo, a gente não vai ter tempo hábil de em uma vida cobrir todo esse grid. Então o random search vai sorteando pontos do grid. Ele sorteia, sorteia, sorteia, sorteia e aí a ideia é que eu cubra um número interessante de possibilidades bem diversas e acabe pegando um bom modelo, ainda que eu não teste exaustivamente todas as possibilidades. Até mesmo porque todas as possibilidades são infinitas. Ou alguma coisa muito perto disso. Praticamente infinitas possibilidades. É um problema que eles chamam de intratável. A gente faz o grid search desse jeito. Ele fica rapidamente intratável. Aí o random search é uma alternativa pra tornar isso viável. E tem outros métodos aí também. Tem um random search ponderado, tem o Bayesian search tem um monte de outros aí. A gente tá passando pelos mais básicos aqui pra dar insumos pra gente poder se desenvolver depois. Além disso. E aí o grid search vale lembrar que a gente fita ele como se fosse um modelinho, como se fosse uma árvore igualzinho. Indica o treino, o teste e tudo. E dentro do grid search a gente vai ter atributos e métodos como o melhor conjunto de hiperparâmetros a melhor classificação e tem o melhor modelo mesmo que a gente usou no num outro código aí que a gente usou grid search. Mas a gente pode pegar o próprio melhor modelo. Bom, Ju, lembrando, eu deixei a lição de casa aqui pra vocês duas do Human Activity Recognition e do a outra menorzinha das gorjetas que é com variável resposta contínua. Tem alguma perguntinha aí? Pro, nenhuma pergunta. Bom, então deixa eu ir fechando aqui mesmo. Bom, então assim, por hoje é só de novo. Só na próxima aula a gente vai falar do Extreme Gradient Boosting que é um algoritmo baseado na ideia de eu pegar o erro do modelo e colocar ele como uma feature numa próxima rodada. E aí a gente vai estar sempre tentando melhorar o erro do modelo sucessivamente a cada interação, a cada passinho. O Boosting ele é isso, a gente vai fazendo uma correção sequencial de erros. Bom, daí o nosso objetivo de hoje era bem esse, a gente falar de Random Forest K-Fold e variável resposta contínua. Acho que deu pra gente cobrir bastante coisa hoje. E ainda sobram uns minutinhos pra tomar uma água aí. Dez minutinhos pelo meu relógio aqui. Olha professor, se o senhor me permitir e se algum de vocês pessoal tiverem dúvidas nesses minutinhos finais da aula, por favor envie aqui pra nós. Eu faço a leitura prontamente ao professor, né? A gente procura selecionar sempre. Aliás, até chegou aqui uma pergunta. Vocês vão mandando, né? Por vezes os nossos professores mediadores respondem pra vocês de imediato e outras vezes eles selecionam pra que eu possa ler pro professor. E chegou uma pergunta aqui do Felipe, de Curitiba. Vou perguntar pra você, professor. O que precisaria modelar com essas bases? Isso, boa pergunta. Na Human Activity Recognition, o objetivo é a gente fazer um algoritmo capaz de identificar qual é a atividade física que o cara tá fazendo baseado no acelerômetro giroscópio dele. Então a gente imagina, por exemplo, uma manipulação da aplicação. Aqueles reloginhos de esporte pra triatleta. O cara tá correndo, o cara tá pedalando e o cara tá nadando durante a prova. Então o reloginho tem que conseguir, tem que ser capaz de identificar qual é a atividade física que ele tá fazendo. E aí, assim, uma coisa geral como ter um celular no bolso, ele consegue fazer coisas como contar o número de passos, dizer se você tá sedentário, se você passou muito tempo sentado. Esse exercício, a gente vai adivinhar só, a cada momento, o que a pessoa tá fazendo. Se ela tá sentada, se ela tá deitada, se ela tá andando, subindo escada, descendo escada e em pé parada. O outro de gorjeta, a gente vai tentar adivinhar o percentual de gorjeta que vai ser paga assim, será que quando o homem paga, ele paga mais gorjeta ou menos gorjeta? Quando a pessoa é fumante, ela paga mais gorjeta ou menos gorjeta? Vou dar um spoiler que se a gente quer ter uma previsão muito boa, dependendo das variáveis que a gente pegar, esse exercício pode ser mais frustrante. Assim, dependendo das variáveis que a gente pegar, pode ser sentir quase que trapaceando, assim, né? Mas é um exercício de variável resposta contínua pra gente se acostumar um pouquinho com as métricas de avaliação e tudo. Pró, não temos mais perguntas. E olha, pessoal, eu vou aproveitar esse minutinho final pra reforçar, então, a importância aí de vocês reforçarem os estudos. Eu sempre brinco aqui com vocês, né? A importância de aula dada, aula estudada. Então, revisem, né? Revisem, né? Todo o conteúdo com calma. A gente sabe que, né? Aula à noite, às vezes a gente também um pouquinho tem a exaustão do dia. E eu parabenizo vocês por estarem aqui, né? Se dedicando, se esforçando. E também, pessoal, peço de desculpa se eventualmente às vezes a gente não consegue, assim que vocês mandam prontamente, que às vezes a gente tá olhando também uma outra demanda, tá? E agradeço a compreensão de vocês, agradeço por vocês estarem aqui, se dedicando nessa aula. Agradeço também toda a equipe que esteja, que esteve aqui prontamente pra auxiliá-los, pra transmitir a nossa aula. E olha, a gente tá sempre aqui, de coração aberto pra auxiliar os alunos da melhor forma. E obrigada, professor. Obrigada pela aula de hoje também. Obrigado, pessoal. Até terça-feira que vem, então, de exercício feito e compartilhando as impressões da experiência aí da semana. E tragam as dúvidas, pessoal. O importante, né? Até, professor, a gente pode deixar combinado aqui. É assim que vocês entrarem na semana que vem. Se tiverem dúvidas, vão relatando as dúvidas de vocês e a gente já faz um bate-papo logo no comecinho. O professor auxilia, né? Da melhor forma pra que a gente vá seguindo com o conteúdo, tá bom? Então, fiquem com Deus. Uma ótima noite e até terça-feira. Obrigada, professor. Obrigada, pessoal. Tchau, tchau. Valeu, gente. Até terça. .