Relatório <11> - <Prática: Predição e a Base de Aprendizado de Máquina (II))>

<Jonas Correia>

1. **[Activity] Linear Regression**
2. **[Activity] Polynomial Regression**
3. **[Activity] Multiple Regression, and Predicting Car**
4. **[Activity] PCA Example with the Iris data set**
5. **Multi-Level Models**

Os modelos multiníveis (também conhecidos como modelos hierárquicos ou de efeitos mistos) são uma abordagem usada para lidar com dados que possuem uma estrutura hierárquica ou agrupada. Isso significa que os dados podem ser organizados em vários níveis de análise, onde os fatores de um nível afetam os de outro, criando interdependências complexas.

Por exemplo, imagine que você está analisando o desempenho de alunos em uma prova. Esses alunos estão agrupados em turmas, que por sua vez estão em escolas, dentro de distritos escolares. Nesse caso, o desempenho individual de um aluno pode ser afetado por fatores específicos de sua própria experiência (nível individual), fatores relacionados à turma em que ele está inserido (nível da turma), fatores da escola (nível da escola) e até fatores maiores, como políticas educacionais de um distrito (nível distrital).

Em um modelo simples, talvez você tentasse prever o desempenho de um aluno apenas com base nas características pessoais dele, como horas de estudo ou QI. Mas um modelo multinível reconhece que existem influências tanto no nível do aluno quanto em níveis mais altos (por exemplo, a qualidade do ensino na escola ou o ambiente familiar) que precisam ser levadas em conta.

6. **Supervised vs. Unsupervised Learning, and Train/Test**

A aprendizagem de máquina envolve algoritmos que aprendem a partir de dados observacionais para fazer previsões. Esse processo pode ser dividido em dois tipos principais: a aprendizagem supervisionada e a não supervisionada. Na primeira, o modelo é treinado com dados rotulados, ou seja, acompanhados de respostas corretas, para que possa fazer previsões sobre novos dados. Já na aprendizagem não supervisionada, o modelo tenta encontrar padrões ou grupos nos dados sem ter acesso a essas respostas previamente. Um exemplo é quando um algoritmo agrupa objetos com base em suas semelhanças, sem uma categorização definida previamente.

Um conceito essencial para avaliar o desempenho de modelos supervisionados é o "Train/Test", onde os dados são divididos em dois grupos:

um para treinar o modelo e outro para testá-lo. Dessa forma, é possível verificar se o modelo consegue generalizar seu aprendizado para novos dados, evitando o problema de overfitting, que ocorre quando o modelo se ajusta muito aos dados de treinamento e perde sua capacidade de fazer previsões precisas em novos dados.

Para tornar a avaliação mais robusta, existe a validação cruzada K-fold, uma técnica que divide os dados em várias partes, treinando e testando o modelo repetidamente em diferentes subconjuntos dos dados. Isso melhora a precisão da avaliação, garantindo que o modelo não esteja se ajustando apenas a um conjunto específico de dados. Assim, a aprendizagem de máquina abrange desde métodos simples de regressão até abordagens mais complexas para assegurar que os modelos possam prever com precisão novos resultados.

7. [Activity] Using Train/Test to Prevent Overfitting a Polynomial Regression

8. Bayesian Methods: Concepts

O classificador de spam em e-mails é uma aplicação interessante de técnicas de aprendizado de máquina, especificamente do método conhecido como Naive Bayes, que se fundamenta no Teorema de Bayes. Essa abordagem permite determinar a probabilidade de um e-mail ser classificado como spam com base em características observadas em mensagens anteriores. O Teorema de Bayes calcula a probabilidade de um evento, considerando a informação prévia disponível.

Para construir um classificador de spam, o modelo é treinado utilizando um conjunto de e-mails já identificados como spam ou não spam. Por exemplo, ao analisar um e-mail que contém a palavra "grátis", o classificador utiliza o Teorema de Bayes para calcular a probabilidade de que esse e-mail seja spam, considerando quantas vezes a palavra aparece em mensagens de spam comparado ao total de e-mails analisados.

No entanto, para o classificador ser eficaz, ele deve considerar não apenas uma única palavra, mas todas as palavras presentes no e-mail. O modelo deve conseguir analisar cada termo e determinar sua contribuição para a probabilidade geral de o e-mail ser spam. Essa análise é facilitada por ferramentas como o Scikit-learn em Python, que permite dividir e processar as palavras eficientemente. A técnica é chamada de "Naive" porque assume que as palavras são independentes entre si, simplificando o cálculo, embora essa suposição nem sempre reflita a realidade.

A implementação do classificador de spam com essas técnicas torna o processo acessível e prático, permitindo filtrar mensagens indesejadas e aprimorar a experiência do usuário na gestão de e-mails. Assim, o uso do Naive Bayes não apenas exemplifica a aplicação de conceitos matemáticos na

tecnologia, mas também mostra como a ciência de dados pode impactar a comunicação digital.

9. [Activity] Implementing a Spam Classifier with Naive Bayes

O K-means é uma técnica de aprendizado de máquina não supervisionado, amplamente utilizada para agrupar dados com base em suas características. O princípio fundamental dessa abordagem é a divisão de um conjunto de dados em K grupos ou aglomerados, onde K é um número definido pelo usuário. O processo envolve a identificação de K centróides, que são os pontos centrais de cada grupo. Cada ponto de dado é associado ao centróide mais próximo, e esse agrupamento é ajustado iterativamente para melhor refletir a distribuição dos dados.

O algoritmo começa escolhendo aleatoriamente K centróides em um gráfico de dispersão. Em seguida, para cada ponto de dado, calcula-se a distância até cada centróide e o ponto é atribuído ao centróide mais próximo. Após essa atribuição, os centróides são recalculados com base nas novas associações de pontos, movendo-os para o centro dos pontos que pertencem a cada grupo. Este processo é repetido até que os centróides não se movam significativamente, indicando que uma convergência foi alcançada.

Embora o K-means seja uma técnica poderosa, existem desafios associados à sua aplicação. Um dos principais é a escolha do valor de K, que não é uma tarefa simples. Uma abordagem comum para determinar K é começar com valores baixos e aumentá-lo gradualmente, observando a redução no erro quadrático até que se alcance um ponto de estagnação, indicando que não se obtém mais informações relevantes ao adicionar mais grupos. Além disso, o algoritmo pode convergir para mínimos locais, dependendo da escolha inicial dos centróides. Por isso, é aconselhável executar o K-means várias vezes com diferentes inicializações e combinar os resultados.

Outro ponto a ser considerado é que o K-means não fornece rótulos ou interpretações explícitas para os agrupamentos formados. Embora o algoritmo consiga identificar relações entre os dados, cabe ao usuário analisar e nomear esses grupos com base nas características observadas. Assim, a interpretação dos resultados pode ser um desafio, mas também uma oportunidade de explorar padrões ocultos nos dados.

O uso de ferramentas como o Scikit-learn facilita a implementação do K-means, permitindo que os usuários explorem essa técnica de maneira acessível e prática, enquanto descobrem agrupamentos e padrões que podem não ser imediatamente evidentes.

10. Measuring Entropy

Entropia, no contexto de ciência de dados, é uma medida que quantifica a desordem ou incerteza em um conjunto de dados. Esse conceito, emprestado da física e termodinâmica, nos ajuda a entender quão uniformes ou variados são os elementos em um conjunto de informações. Imagine que você possui um conjunto de animais classificados por espécie. Se todos os animais forem da mesma espécie, a entropia será muito baixa, pois há pouca variação — todos são iguais. No entanto, se cada animal for de uma espécie diferente, a entropia será alta, já que há muita diversidade, ou desordem, no conjunto.

Basicamente, a entropia nos diz o quanto um conjunto de dados é homogêneo ou heterogêneo. Quando a entropia é zero, significa que todas as classes são iguais. Se a entropia é alta, as classes são muito diferentes entre si. Em um nível intuitivo, o conceito é simples: a entropia descreve quão uniformes ou diversas são as coisas em um conjunto de dados.

Matematicamente, a entropia é calculada considerando a proporção de elementos de cada classe em relação ao conjunto total de dados. Para cada classe, você calcula a probabilidade (P) de um dado pertencer àquela classe e, em seguida, usa essa probabilidade para calcular a contribuição de cada classe para a entropia total. A fórmula envolve o produto de P pelo seu logaritmo natural, e o resultado é somado para todas as classes. Se a proporção de uma classe for zero ou cem por cento, sua contribuição para a entropia é nula, pois todos pertencem àquela classe ou nenhum pertence.

11. [Activity] WINDOWS: Installing Graphviz

12. Decision Trees: Concepts

Uma árvore de decisão é uma ferramenta amplamente utilizada no aprendizado de máquina, que funciona como um fluxograma para auxiliar na tomada de decisões com base em vários atributos. Seu objetivo é prever resultados futuros com base em dados históricos. Ao fornecer ao algoritmo um conjunto de dados de treinamento, ele pode construir uma árvore que segmenta os dados em diferentes níveis, cada um representando uma decisão com base em um atributo específico. Esses atributos podem ser fatores como a umidade, a temperatura ou a presença de sol, no exemplo de decidir se uma pessoa deve ou não sair para brincar.

As árvores de decisão utilizam o conceito de entropia para orientar suas escolhas. A entropia é uma medida de desordem ou incerteza nos dados, e o objetivo do algoritmo é escolher atributos que a minimizem em cada divisão. Ao longo do processo, o algoritmo analisa qual atributo melhor separa os dados de

forma que, em cada etapa, a árvore se aproxime de uma decisão mais clara, como "sim" ou "não" em relação à questão sendo analisada.

Um exemplo prático mencionado foi o uso de árvores de decisão para filtrar currículos, avaliando características como a experiência de trabalho, nível de educação, estágio e se o candidato frequentou uma universidade de prestígio. O modelo, ao ser treinado com dados de candidatos anteriores e os resultados de suas contratações, pode construir um fluxograma que ajudará a determinar se um novo candidato tem uma alta probabilidade de ser contratado.

O algoritmo responsável por construir essas árvores é chamado ID3, que segue uma abordagem gananciosa, ou seja, em cada etapa ele escolhe a melhor opção para minimizar a entropia naquele momento. No entanto, um dos desafios das árvores de decisão é o overfitting, um fenômeno no qual o modelo se ajusta muito bem aos dados de treinamento, mas não se generaliza bem para novos dados. Para resolver esse problema, uma técnica chamada floresta aleatória é frequentemente utilizada. As florestas aleatórias criam várias árvores de decisão, cada uma treinada em uma amostra aleatória dos dados e, em seguida, todas as árvores votam no resultado final. Esse processo de votação ajuda a reduzir o risco de overfitting e torna o modelo mais robusto.

13. [Activity] Decision Trees: Predicting Hiring Decisions

14. Ensemble Learning

O aprendizado em conjunto é uma abordagem poderosa na aprendizagem de máquina que envolve a combinação de vários modelos para melhorar a precisão e a robustez das previsões. Em vez de confiar em um único modelo, utilizam-se vários modelos que trabalham juntos para produzir um resultado final mais eficaz. Essa técnica, como vimos no caso da Random Forest, utiliza múltiplas árvores de decisão, onde cada uma é treinada em subamostras dos dados e em diferentes conjuntos de atributos. Ao final, todas votam para decidir a classificação.

Existem diferentes métodos de aprendizado em conjunto. O ensacamento (ou bootstrap aggregating) é uma técnica onde múltiplos modelos são treinados com amostras aleatórias dos dados. No caso da Random Forest, cada árvore de decisão é construída com uma amostra aleatória dos dados de treinamento e, em seguida, todas elas contribuem para o resultado final. Essa abordagem ajuda a evitar o overfitting, proporcionando previsões mais generalizáveis.

Outro método de aprendizado em conjunto é o reforço, que se diferencia do ensacamento. No reforço, cada novo modelo tenta corrigir os erros cometidos pelo modelo anterior, concentrando-se nas áreas onde a classificação falhou. A ideia é aumentar a precisão progressivamente, construindo modelos que aprimoram os erros anteriores.

Um terceiro método, conhecido como balde de modelos, permite o uso de diferentes tipos de modelos (como k-means, árvores de decisão e regressão) para fazer previsões. Nesse caso, os modelos "competem" entre si, e o modelo com o melhor desempenho é escolhido para fazer a previsão final. Já no empilhamento, os resultados de vários modelos são combinados de alguma forma para melhorar o resultado final, em vez de escolher apenas um modelo vencedor.

O aprendizado em conjunto mostrou-se extremamente eficaz em diversas aplicações práticas. Um exemplo famoso é o concurso da Netflix, onde pesquisadores utilizaram várias abordagens de conjunto para superar o algoritmo de recomendação de filmes da empresa, conseguindo resultados significativamente melhores ao combinar diferentes algoritmos.

Embora existam técnicas avançadas de aprendizado em conjunto, como o Classificador Ideal Bayes e o Bayesian Model Combination, essas abordagens são geralmente complexas e pouco práticas. Muitas vezes, técnicas mais simples, como o ensacamento e o empilhamento, são suficientes para alcançar excelentes resultados. Por isso, a recomendação é começar com as soluções mais simples e, apenas se necessário, explorar alternativas mais sofisticadas.

15. [Activity] XGBoost

16. Support Vector Machines (SVM) Overview

As Máquinas de Vetores de Suporte (SVM) são uma técnica poderosa para classificar e agrupar dados em dimensões elevadas, ou seja, quando há muitas características ou atributos envolvidos. Elas são particularmente eficazes quando estamos lidando com dados complexos e várias variáveis ao mesmo tempo. O objetivo principal de uma SVM é encontrar um "hiperplano" que melhor separe os dados em diferentes classes. A matemática por trás das SVMs pode ser complexa, mas ferramentas como o scikit-learn facilitam a implementação prática sem a necessidade de entrar nos detalhes técnicos.

O conceito fundamental das SVMs gira em torno dos "vetores de suporte", que são pontos de dados críticos que definem o hiperplano de separação. A técnica utiliza um "truque do kernel" para lidar com dados não-linearmente separáveis, permitindo que sejam encontrados hiperplanos em dimensões superiores. Isso significa que, em vez de simplesmente dividir dados em duas dimensões, uma SVM pode trabalhar em espaços multidimensionais, o que a torna muito útil para dados complexos.

Os kernels (núcleos) são funções que ajudam a projetar os dados em espaços de maior dimensão. Existem diferentes tipos de kernels, como o linear, polinomial e radial, e a escolha do kernel adequado é importante para a eficácia do modelo. Cada kernel tem um custo computacional diferente, então, escolher o correto é essencial para balancear precisão e eficiência.

Outro ponto importante sobre as SVMs é que elas são uma técnica de aprendizagem supervisionada, o que significa que precisam de um conjunto de dados de treinamento com rótulos corretos para aprender. Isso as diferença de técnicas não supervisionadas, como o k-means, que não exige rótulos durante o treinamento.

Um exemplo clássico da aplicação de SVMs é a classificação do conjunto de dados Iris, que contém medições de flores de diferentes espécies. Com base no comprimento e largura das pétalas e sépalas, as SVMs podem prever a espécie da flor. Dependendo do kernel utilizado, os resultados podem variar, e há um equilíbrio entre a complexidade do modelo e sua capacidade de generalização.

Ao usar SVMs, é crucial realizar validação cruzada para encontrar os melhores parâmetros e evitar o sobreajuste (quando o modelo se ajusta demais aos dados de treinamento e não generaliza bem para novos dados). Portanto, o teste e ajuste dos parâmetros é uma parte essencial para garantir que o modelo tenha bom desempenho em dados não vistos

17. [Activity] Using SVM to cluster people using scikit-learn