Atividade de Machine Learning
Ciência da Computação 5º período
Alunos:
- **Eric Freitas Avelino**
- **Ângelo Garcia Pereira**
- **Giovanna Gallucio Crisostomo**

Pesquisa de Algoritmos de classificação

Regressão Logística

A regressão logística é um algoritmo super potente de classificação que nos permite saber, por exemplo, se um cliente é bom ou ruim. É importante destacar que por mais que o nome seja “regressão logística” ele não é um algoritmo de regressão e sim de classificação. na regressão logística, ao invés de determinar um valor binário para a classe (0 ou 1, maçã ou banana, fraude ou não), ela retorna à probabilidade de um evento ocorrer.
Ou seja, a probabilidade de termos um cliente bom ou ruim, a probabilidade de ocorrerem fraudes, a probabilidade de um cliente sair da empresa, etc.

Naive Bayes

O classificador Naive Bayes é um algoritmo que se baseia nas descobertas de Thomas Bayes para realizar predições em aprendizagem de máquina. O termo “naive” (ingênuo) diz respeito à forma como o algoritmo analisa as características de uma base de dados: ele assume que as features são independentes entre si.
Além disso, ele também assume que as variáveis features são todas igualmente importantes para o resultado. Em cenários em que isso não ocorre, essa técnica deixa de ser a opção ideal. Discutiremos adiante sobre as aplicações.

Como Bayes é um nome famoso na estatística, é fácil concluir que o seu algoritmo tem uma forte base dessa área, reforçando a relação entre estatística e inteligência artificial.
Inclusive, o seu funcionamento pode ser facilmente descrito em termos estatísticos: para calcular a predição, o algoritmo define, primeiramente, uma tabela de probabilidades, em que consta a frequência dos preditores com relação às variáveis de saída. Então, o cálculo final leva em conta a probabilidade maior para oferecer uma solução.

Árvore de decisão

Uma árvore de decisão é um algoritmo de aprendizado de máquina supervisionado que é utilizado para classificação e para regressão. Isto é, pode ser usado para prever categorias discretas (sim ou não, por exemplo) e para prever valores numéricos (o valor do lucro em reais).

Assim como um fluxograma, a árvore de decisão estabelece nós (decision nodes) que se relacionam entre si por uma hierarquia. Existe o nó-raiz (root node), que é o mais importante, e os nós-folha (leaf nodes), que são os resultados finais. No contexto de machine learning, o raiz é um dos atributos da base de dados e o nó-folha é a classe ou o valor que será gerado como resposta.

Na ligação entre nós, temos regras de “se-então”. Ao chegar em um nó A, o algoritmo se pergunta acerca de uma regra, uma condição, como “se a característica X do registro analisado é menor do que 15?”. Se for menor, então ele vai para um lado da árvore; se for maior, então ele vai para outro. No próximo nó, segue a mesma lógica.

É um algoritmo que segue o que chamamos de “recursivo” em computação. Ou seja, ele repete o mesmo padrão sempre na medida em que vai entrando em novos níveis de profundidade. É como se uma função chamasse a ela mesma como uma segunda função para uma execução paralela, da qual a primeira função depende para gerar sua resposta.

Random Forest

Random forest é um algoritmo de aprendizado de máquina amplamente utilizado, registrado por Leo Breiman e Adele Cutler, que combina a saída de múltiplas decision trees para alcançar um único resultado. Sua facilidade de uso e flexibilidade impulsionaram sua adoção, pois lida com problemas de classificação e regression. Para entender o algoritmo RandomForest, precisamos primeiramente conhecer os métodos ensemble, dos quais ele faz parte.

Estes métodos são construídos da mesma forma que algoritmos mais básicos, como regressão linear, árvore de decisão ou knn, por exemplo, mas possuem uma característica principal que os diferenciam, a combinação de diferentes modelos para se obter um único resultado. Essa característica torna esses algoritmos mais robustos e complexos, levando a um maior custo computacional que costuma ser acompanhado de melhores resultados.

Normalmente na criação de um modelo, escolhemos o algoritmo que apresenta o melhor desempenho para os dados em questão. Podemos testar diferentes configurações deste algoritmo escolhido, gerando assim diferentes modelos, mas no fim do processo de machine learning, escolhemos apenas um.
Com um método ensemble serão criados vários modelos diferentes a partir de um algoritmo, mas não escolhemos apenas um para utilização final, e sim todos.

No algoritmo RandomForest serão criadas várias árvores de decisão, sendo este conhecimento fundamental para o entendimento do algoritmo.

KNN
O algoritmo KNN permite com que você faça previsões de dados com base nos K vizinhos mais próximos a esse ponto
Como esse algoritmo faz previsão é basicamente medir a distância dos pontos ao redor dele.
O resultado é definido como da mesma classe da maioria dos pontos mais próximos a ele e somos nós que escolhemos a quantidade de pontos que serão analisados.

Máquina de Vetores de Suporte
Uma máquina de vetores de suporte (SVM) é um algoritmo supervisionado de aprendizado de máquina que classifica dados encontrando uma linha ou hiperplano ótimo que maximiza a distância entre cada classe em um espaço N-dimensional. As SVMs são comumente usadas em problemas de classificação. Eles distinguem duas classes encontrando o hiperplano ideal que maximiza a margem entre os pontos de dados mais próximos de classes opostas. O número de atributos dos dados de entrada determina se o hiperplano é uma linha em um espaço 2D ou um plano em um espaço n-dimensional. Como vários hiperplanos podem ser encontrados para diferenciar as classes, maximizar a margem entre os pontos permite que o algoritmo encontre o melhor limite de decisão entre as classes.

● SVMs lineares

As SVMs lineares são usadas com dados linearmente separáveis, o que significa que os dados não precisam passar por transformações para serem separados em diferentes classes.

Matematicamente, esse hiperplano separador pode ser representado como:
$wx + b = 0$
em que w é o vetor de peso, x é o vetor de entrada e b é o termo de viés.
Há duas abordagens para calcular a margem, ou a distância máxima entre as classes, que são a classificação com margem rígida e a classificação com margem flexível.

Isso é representado pela fórmula,
$(wxj + b)\, yj \geq a$,

● SVMs não lineares
Grande parte dos dados em cenários do mundo real não são separáveis de forma linear, e é aí que as SVMs não lineares entram em ação. A fim de tornar os dados separáveis de forma linear, métodos de pré-processamento são aplicados aos dados de treinamento para transformá-los em um espaço de atributos de maior dimensão. Dito isso, os espaços dimensionais superiores são capazes de criar mais

complexidade, aumentando o risco de sobreajuste dos dados e aumentando a exigência computacional. O "truque do kernel" ajuda a reduzir parte dessa complexidade, tornando a computação mais eficiente, e faz isso substituindo os cálculos de produto escalar por uma função de kernel equivalente4.
Existem diversos tipos de kernels que podem ser aplicados para classificar dados. Algumas funções de kernel populares incluem:
• Kernel polinomial
• Kernel de função de base radial (também conhecido como kernel gaussiano ou RBF)
• Kernel Sigmoide

● Regressão por vetores de suporte (SVR)

A regressão por vetores de suporte (SVR) é uma extensão das SVMs, aplicada a problemas de regressão (ou seja, o resultado é contínuo). De modo semelhante às SVMs lineares, o SVR encontra um hiperplano com a margem máxima entre os pontos de dados e é normalmente usado para a previsão de séries temporais.
O SVR difere da regressão linear porque é necessário especificar a relação que se deseja entender entre as variáveis independentes e dependentes. Compreender as relações entre variáveis e suas direções é valioso ao usar a regressão linear. Isso não é necessário para os SVRs, pois eles determinam essas relações automaticamente.

XGBoost
O XGBoost (eXtreme Gradient Boosting) é uma biblioteca de aprendizado de máquina distribuída e de código aberto que utiliza árvores de decisão com reforço gradativo, um algoritmo de aprendizado supervisionado que faz uso do gradiente descendente. É conhecido por sua velocidade, eficiência e capacidade de escalar bem com grandes conjuntos de dados.

LightGBM
O LightGBM é um framework de aprendizado de máquina que utiliza árvores de decisão como base para realizar tarefas de classificação e regressão. Sua eficiência computacional alta o torna ideal para lidar com grandes conjuntos de dados. Além disso, o LightGBM suporta paralelismo, o que permite que ele seja executado em múltiplos processadores, reduzindo o tempo de treinamento.

Uma das principais vantagens do LightGBM é sua capacidade de lidar com dados esparsos, ou seja, dados que contêm muitos zeros. Isso é especialmente útil em problemas de classificação e regressão, onde os dados podem ser esparsos. Além disso, o LightGBM também suporta a utilização de recursos de aprendizado de máquina, como o boosting, que pode ajudar a melhorar a precisão do modelo.

No entanto, o LightGBM também tem algumas desvantagens. Uma delas é que ele pode ser sensível a hiperparâmetros, o que pode exigir uma ajuste cuidadoso para obter os melhores resultados. Além disso, o LightGBM também pode ser difícil de interpretar, especialmente para modelos complexos.

Em resumo, o LightGBM é uma ferramenta poderosa para problemas de classificação e regressão, especialmente quando se lida com grandes conjuntos de dados. No entanto, é importante ter cuidado ao ajustar os hiperparâmetros e interpretar os resultados.

CatBoost

O CatBoost é uma biblioteca de aprendizado de máquina que utiliza árvores de decisão como base e é projetada para lidar com dados categóricos de forma eficiente. Sua capacidade de lidar com dados categóricos sem a necessidade de pré-processamento os torna ideais para problemas de classificação e regressão.

Uma das principais vantagens do CatBoost é sua capacidade de lidar com dados categóricos de forma eficiente. Isso é especialmente útil em problemas de classificação e regressão, onde os dados podem conter muitas variáveis categóricas. Além disso, o CatBoost também suporta a utilização de recursos de aprendizado de máquina, como o boosting, que pode ajudar a melhorar a precisão do modelo. Outra vantagem do CatBoost é sua facilidade de uso. Ele é fácil de instalar e utilizar, e não requer conhecimento avançado de programação. Além disso, o CatBoost também é rápido e eficiente, o que o torna ideal para problemas de grande escala.

No entanto, o CatBoost também tem algumas desvantagens. Uma delas é que ele pode ser mais lento do que o LightGBM para problemas de grande escala. Além disso, o CatBoost também pode não ser tão flexível quanto outras técnicas de aprendizado de máquina.

Em resumo, o CatBoost é uma ferramenta poderosa para problemas de classificação e regressão, especialmente quando se lida com dados categóricos. Sua facilidade de uso e eficiência o tornam ideal para problemas de grande escala.

Redes Neurais Artificiais (RNA)

As Redes Neurais Artificiais (RNA) são modelos de aprendizado de máquina inspirados na estrutura e funcionamento do cérebro humano. Sua capacidade de aprender padrões complexos em dados as torna ideais para problemas de classificação e regressão.

Uma das principais vantagens das RNA é sua capacidade de aprender padrões complexos em dados. Isso é especialmente útil em problemas de classificação e regressão, onde os dados podem conter muitas variáveis e padrões complexos.

Além disso, as RNA também podem ser utilizadas em uma variedade de tarefas, desde a análise de imagens até a previsão de séries temporais.

Outra vantagem das RNA é sua capacidade de aprender a partir de dados não estruturados. Isso é especialmente útil em problemas de classificação e regressão, onde os dados podem ser não estruturados ou contener muitas variáveis. No entanto, as RNA também têm algumas desvantagens. Uma delas é que elas podem ser difíceis de treinar e ajustar. Isso é especialmente verdadeiro para problemas complexos, onde os dados podem conter muitas variáveis e padrões complexos. Além disso, as RNA também podem requerer grandes conjuntos de dados para treinamento, o que pode ser um desafio em problemas onde os dados são escassos. Além disso, as RNA também podem ser computacionalmente intensivas e difíceis de interpretar.

No entanto, as RNA são uma ferramenta poderosa para problemas de classificação e regressão, especialmente quando se lida com dados complexos e não estruturados. Sua capacidade de aprender padrões complexos em dados as torna ideais para problemas de análise de imagens, processamento de linguagem natural e previsão de séries temporais.

Em resumo, as RNA são uma ferramenta poderosa para problemas de classificação e regressão, mas requerem cuidado ao treinar e ajustar, e podem ser computacionalmente intensivas e difíceis de interpretar.