EDGARD JOSEPH KIRIYAMA

DATA ANALYTICS

MACHINE LEARNING COM PYTHON

AULA 06

SUMÁRIO

O QUE VEM POR AÍ?

Olá, jovem analytic!

Você está na terceira disciplina do curso e aqui começa mais um ponto importantíssimo da jornada em que você está trilhando.

Na aula passada falamos sobre modelos de Machine Learning com algoritmos de regressão, suas peculiaridades e o que devemos nos atentar para validar nossos modelos.

Chegamos em outro momento importante de nossa jornada! Vamos falar sobre modelos de Machine Learning com algoritmos de classificação.

Este é o momento de aprimorarmos e elevarmos o nosso nível! Vamos te mostrar o que são modelos de ML de classificação, quais são os modelos mais utilizados e como avaliar se o modelo está realizando as predições de forma correta.

Vamos entender como este conhecimento pode nos ajudar no nosso dia a dia! Bora lá?

CONHEÇA SOBRE O ASSUNTO

O Machine Learing (Aprendizado de Máquina), é um método de análise de dados que automatiza a construção de modelos analíticos.

Relembrando o que tivemos nas aulas passadas, os modelos de Machine Learning (ML) são classificados da seguinte forma:

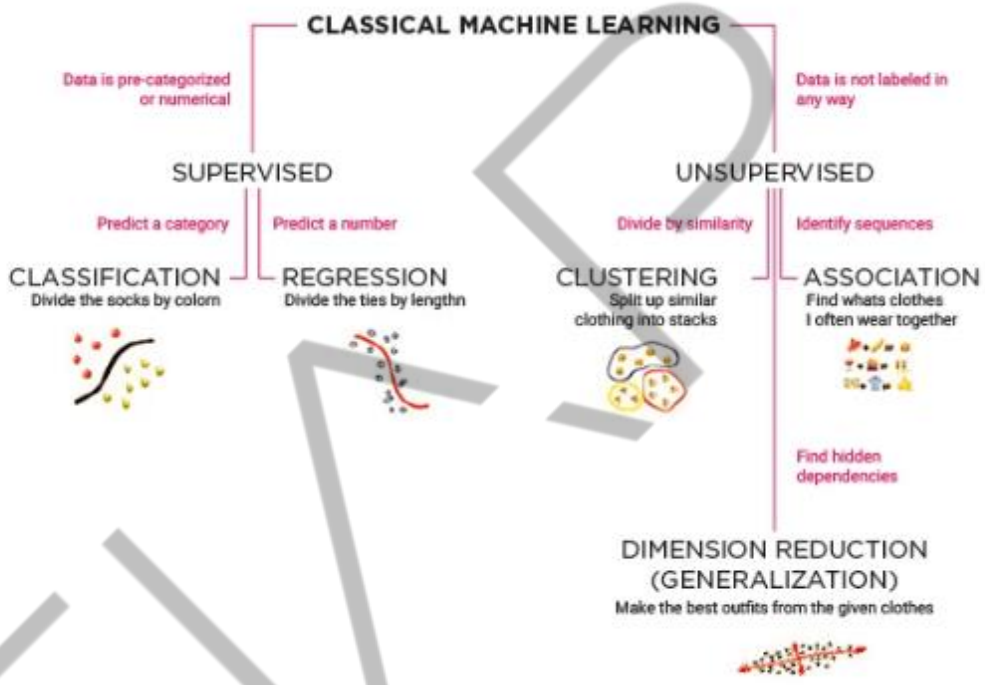

Figura 1 - Estrutura dos modelos clássicos de Machine Learning
Fonte: https://www.datageeks.com.br/machine-learning/ (2019)

Conforme citamos na aula anterior sobre regressão, o algoritmo supervisionado elabora regras que podem fornecer a saída esperada para as entradas fornecidas. Essa capacidade de tomada de decisão permite que o algoritmo preveja novas entradas e tome decisões. Quando novos dados são encontrados, os dados e regras existentes são utilizados para entender e tomar decisões sobre eles. A fórmula é usada para prever variáveis dependentes (o que você deseja medir) a partir de variáveis independentes (do que você acha que sua medida de destino depende). Os valores previstos são contínuos. A regressão fornece resultados contínuos.

Os modelos supervisionados de classificação são uma técnica comum de aprendizado de máquina usada para prever a classe de um conjunto de dados com base em características ou atributos conhecidos. Esses modelos são

"supervisionados", porque o processo de treinamento é orientado por um conjunto de dados rotulados, em que cada exemplo de treinamento é acompanhado por uma classe conhecida.

Estes modelos são comumente usados em uma ampla variedade de aplicações, como reconhecimento de imagens, detecção de spam de e-mails, previsão de fraudes em cartões de crédito e diagnóstico médico. Eles são uma ferramenta valiosa para organizações que desejam automatizar processos e tomar decisões mais informadas com base em dados.

Os modelos de classificação possuem algumas limitações que podem impactar a sua eficácia e precisão. Algumas dessas limitações incluem:

- Dados desbalanceados: quando uma classe é muito mais frequente que as outras no conjunto de dados, o modelo tende a dar mais importância a essa classe, o que pode levar a uma classificação incorreta das classes minoritárias.
- Dados ruidosos: dados com ruído podem levar o modelo a aprender padrões equivocados ou irrelevantes, prejudicando a sua precisão.
- Falta de dados representativos: se os dados utilizados para treinar o modelo não representam bem a população em que o modelo será aplicado, ele pode não ser capaz de generalizar adequadamente.
- Overfitting: o overfitting ocorre quando o modelo se ajusta demais aos dados de treinamento e acaba perdendo a capacidade de generalização para novos dados.
- Underfitting: o underfitting ocorre quando o modelo é muito simples e não consegue capturar a complexidade dos dados de treinamento, levando a uma baixa precisão de classificação.
- Sensibilidade a parâmetros: a escolha inadequada dos parâmetros do modelo pode levar a uma baixa precisão de classificação.
- Dependência de características relevantes: o modelo pode ser altamente sensível a certas características relevantes para a classificação, e caso estas não estejam presentes ou estejam mal definidas, a precisão da classificação pode ser prejudicada.

- Conflitos entre classes: em alguns casos, as fronteiras entre as classes podem não ser claramente definidas, levando a uma classificação ambígua ou incorreta.

Regressão Logística

A regressão logística é uma técnica de modelagem estatística amplamente utilizada para prever a probabilidade de ocorrência de um evento binário, como "sim" ou "não", "verdadeiro" ou "falso", "1" ou "0". É uma forma de análise de regressão que permite a inclusão de variáveis independentes categóricas e contínuas para prever a probabilidade de um evento ocorrer.

O algoritmo é particularmente útil em problemas de classificação binária, como prever se um cliente comprará ou não um produto ou se um paciente tem ou não uma doença. O modelo de regressão logística calcula a probabilidade de um evento ocorrer e, em seguida, usa uma função logística para transformar essa probabilidade em uma previsão binária.

Os principais benefícios da regressão logística são a facilidade de interpretação dos resultados e a capacidade de lidar com variáveis categóricas e contínuas. Além disso, ela é robusta e pode lidar com dados ausentes ou valores extremos.

No entanto, a regressão logística também tem algumas limitações. É uma técnica paramétrica, o que significa que é necessário especificar um modelo matemático subjacente e seus parâmetros. Além disso, ela pressupõe uma relação linear entre as variáveis independentes e dependentes, o que pode não ser apropriado em todos os casos.

Para avaliar se o modelo de regressão logística está realizando as predições corretamente e tendo uma boa performance, são realizadas as seguintes avaliações:

- Acurácia;
- Ajuste do modelo;
- Seleção de variáveis;
- Multicolinearidade;
- Validação cruzada;

- Ajuste de hiperparâmetros.

A biblioteca mais famosa que auxilia na construção do modelo de Regressão Logística:

- Scikit-Learn: sklearn.linear_model (LogisticRegression)

Naive Bayes

O Naive Bayes é um algoritmo de classificação amplamente utilizado em aprendizado de máquina e mineração de dados. Ele é baseado no Teorema de Bayes, que afirma que a probabilidade de uma hipótese ser verdadeira pode ser calculada a partir da probabilidade de uma evidência, dada a hipótese e da probabilidade da hipótese em si.

O algoritmo assume que todas as variáveis são independentes entre si, o que significa que a presença ou ausência de uma característica não afeta a presença ou ausência de outras características. Isso torna o algoritmo muito rápido e eficiente em grandes conjuntos de dados, mas também pode afetar a precisão do modelo em casos onde as variáveis não são completamente independentes.

É amplamente utilizado em tarefas de classificação, como classificação de e-mails como spam ou não spam, identificação de sentimentos em texto, classificação de notícias, entre outros. Ele funciona comparando a probabilidade de cada classe para uma determinada entrada, calculando a probabilidade condicional para cada característica dada a classe e, em seguida, multiplicando-as para obter a probabilidade total.

Mesmo que o modelo seja um algoritmo simples e rápido, é importante lembrar que ele é apenas uma ferramenta e pode não ser adequado para todas as situações. É sempre importante avaliar os resultados do modelo e compará-los com outras técnicas de classificação para determinar a melhor abordagem para uma determinada tarefa.

Para avaliar o modelo, é importante:

- Acurácia;
- Matriz de Confusão;

- Sensibilidade e Especificidade;
- Curva ROC;
- Validação Cruzada;
- F-score.

A biblioteca mais famosa que auxilia na construção do modelo de Naive Bayes:

- Sckiti-learn sklearn.naive_bayes (GaussianNB)

Suport Vector Machine - SVM

SVM (Support Vector Machines) é um algoritmo de aprendizado de máquina supervisionado, que é usado principalmente para problemas de classificação e regressão. Ele foi proposto pela primeira vez em 1992 por Vladimir Vapnik e seus colegas.

O SVM funciona encontrando o hiperplano de separação que maximiza a distância entre as classes. O hiperplano de separação é escolhido de tal forma que ele divide o espaço de características em duas regiões, uma para cada classe, de forma que a distância entre o hiperplano e os pontos de dados mais próximos de cada classe, chamados vetores de suporte, seja maximizada.

Este algoritmo é muito interessante, pois ele pode lidar com dados não linearmente separáveis usando o Kernel, que mapeia os dados em um espaço de recursos de dimensão superior, onde é mais provável que eles sejam separáveis linearmente. Esses truques de Kernel permitem que o SVM modele relações complexas entre os dados.

Os modelos de SVM têm sido amplamente utilizados em problemas de classificação, como reconhecimento de imagens, detecção de spam e diagnóstico médico. Eles também têm sido aplicados em problemas de regressão, como previsão de preços de ações e previsão de valores de propriedades imobiliárias.

Uma das principais vantagens do SVM é que ele é relativamente fácil de interpretar e entender. Ele também tem um bom desempenho em problemas com muitas características e é resistente a overfitting. No entanto, um dos principais

desafios do SVM é escolher o kernel correto e seus parâmetros para um conjunto de dados específico.

Uma curiosidade: ele é um algoritmo amado por boa parte dos(as) analistas e cientistas de dados!

Para avaliarmos se o modelo classificou de forma correta, devemos verificar:

- Métricas de desempenho como acurácia, precisão, recall, F1-score, área sob a curva ROC (AUC-ROC;
- Validação cruzada;
- Seleção de hiperparâmetros;
- Overfitting.

A biblioteca mais famosa que auxilia na construção do modelo de SVM:

- Sckiti-learn: sklearn.svm (SVC)

Árvores de decisão

Árvore de decisão é uma técnica de aprendizado de máquina supervisionado, usada para classificar dados em categorias. Essa técnica cria uma estrutura em forma de árvore, onde cada nó representa uma decisão baseada em uma característica dos dados. A partir da raiz da árvore, os dados são divididos em subconjuntos sucessivos com base nas características escolhidas, até que cada subconjunto contenha apenas uma categoria.

Durante o processo de construção da árvore, o algoritmo busca identificar as características que melhor distinguem as categorias, permitindo que a árvore seja construída de forma a minimizar a quantidade de erros de classificação. Após a construção da árvore, novos dados podem ser classificados, seguindo o caminho percorrido na árvore a partir das suas características.

A árvore de decisão é uma técnica popular devido à sua simplicidade e facilidade de interpretação. A estrutura em forma de árvore é facilmente visualizável e compreensível, permitindo que os usuários possam entender as decisões tomadas pelo algoritmo e validar a qualidade da classificação. Além disso, a árvore de decisão

é capaz de lidar com dados categóricos e numéricos, tornando-se uma opção versátil para problemas de classificação em diferentes áreas.

Algumas das principais métricas a serem consideradas ao avaliar um modelo de Árvore de Decisão incluem:

- Acurácia;
- Precisão e Recall;
- Matriz de Confusão;
- Overfitting;
- Variedade de características.

A biblioteca mais famosa que auxilia na construção do modelo de Árvore de Decisão:

- Sklearn: sklearn.tree (DecisionTreeClassifier)

Dica de leitura

Que tal dar uma lida no artigo sobre Modelos de classificação no blog do medium? Aqui, o autor Daniel Keras faz uma boa visão e exemplos práticos de ML supervisionado de classificação:

Artigo sobre modelos de classificação.

HANDS ON

Agora, chegou o momento de ver, na prática, como começar a importar os nossos dados e trabalhar com eles via programação. A ideia é não se limitar apenas ao código explícito no hands on, então é sempre bom procurar a documentação das bibliotecas, explorar novas funcionalidades e muito mais!

O QUE VOCÊ VIU NESTA AULA?

Introdução a Modelos de Classificação; aplicação de Machine Learning de Classificação; tipo de Modelos de Classificação.

Daqui em diante, é importante que você replique os conhecimentos adquiridos para fortalecer mais suas bases e conhecimentos.

IMPORTANTE: não esqueça de praticar com o desafio da disciplina, para que assim você possa aprimorar os seus conhecimentos!

Você não está sozinho(a) nesta jornada! Te esperamos no Discord e nas lives com os(as) professores(as) especialistas, onde você poderá tirar dúvidas, compartilhar conhecimentos e estabelecer conexões!

REFERÊNCIAS

DOCUMENTAÇÃO scikit-learng. **Disponível em: <https://scikit-learn.org/stable/modules/classes.html >.** Acesso em: 05 mai 2023.

GRUS, Joel. **Data Science do Zero**. Rio de Janeiro: Alta Books Editora, 2016.

HARRISON, Matt. **Machine Learning**: guia de referência rápida - trabalhando com dados estruturados em python. São Paulo: O'Reilly Media, 2019.

PALAVRAS-CHAVE

Palavras-chave: Python. Machine Learning, Modelos de Classificação.