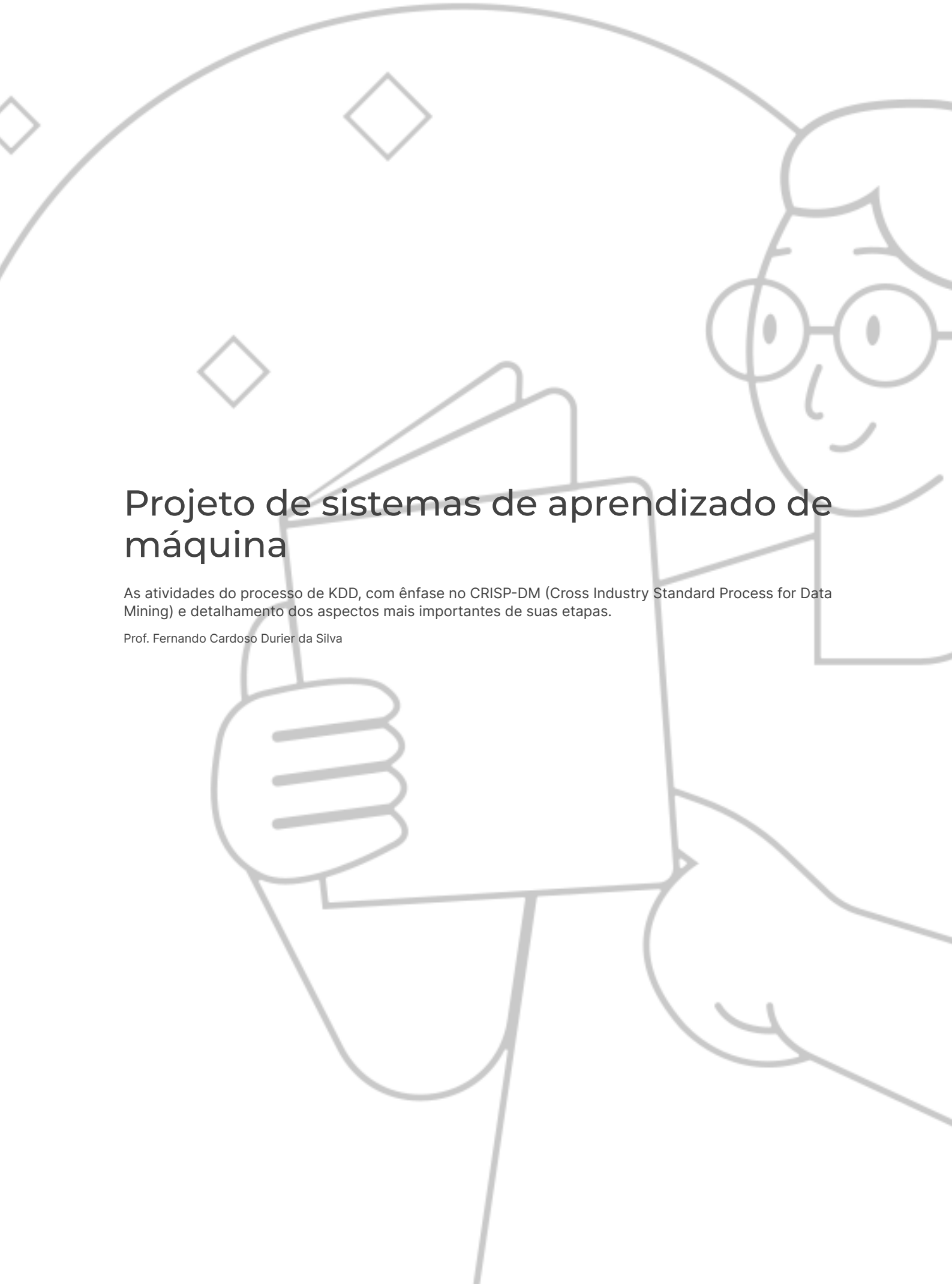


Projeto de sistemas de aprendizado de máquina

As atividades do processo de KDD, com ênfase no CRISP-DM (Cross Industry Standard Process for Data Mining) e detalhamento dos aspectos mais importantes de suas etapas.

Prof. Fernando Cardoso Durier da Silva

1. Itens iniciais

Propósito

Compreender conceitos como modelagem de problemas de aprendizado de máquina, preparação de dados, escolha de algoritmos mediante problemas, enquadramento de problemas com os dados, tratamento de dados irregulares e métricas adequadas, bem como aplicá-los por meio de análise exploratória, construção de processo de experimentação e construção do processo de monitoramento da performance.

Preparação

Para acompanhar as demonstrações deste conteúdo, é necessário ter o Python — na versão 3.7 ou superior — instalado em sua máquina, com as bibliotecas de aprendizado de máquina utilizadas nos exemplos instaladas através do comando pip install.

Objetivos

- Descrever o entendimento do negócio e dos dados.
- Descrever a preparação dos dados.
- Descrever a modelagem e o treinamento de modelos.
- Descrever a avaliação de resultados e a implantação do sistema.

Introdução

Neste conteúdo, compreenderemos o processo de projeto de sistemas de aprendizado de máquina, que se confunde com o processo de KDD (Descoberta de Conhecimento em Bases de Dados: do inglês *Knowledge Discovery in Databases*), detalhando suas principais atividades, entre elas a coleta de dados, a limpeza e o pré-processamento, a preparação dos dados para o treinamento de modelos, e a fase de testes e avaliações.

Entenderemos os diferentes tipos de coleta de dados, quais as técnicas empregadas no pré-processamento, a divisão do conjunto de dados para treinar nossos modelos e, finalmente, quais métricas de monitoramento são mais utilizadas.

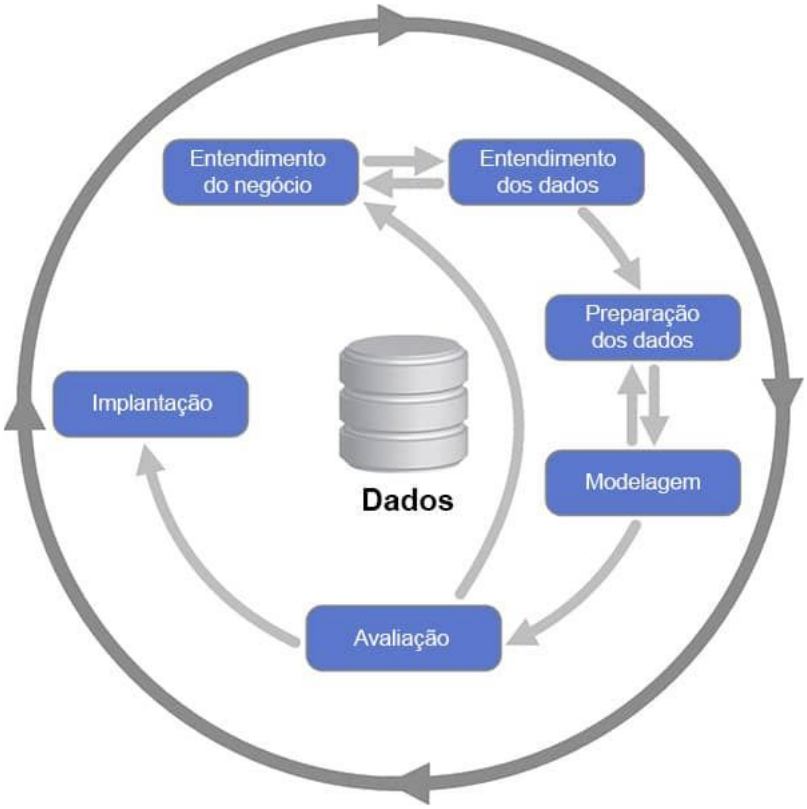
Mapearemos essas atividades para as etapas do CRISP-DM(Do inglês *Cross Industry Standard Process for Data Mining*), um processo consagrado na prática do mercado de KDD, Data Mining e Machine Learning e que consiste em seis etapas:

- Entendimento do negócio (*Business Understanding*)
- Entendimento dos dados (*Data Understanding*)
- Preparação dos dados (*Data Preparation*)

- Modelagem (*Modeling*)
- Avaliação (*Evaluation*)
- Implantação (*Deployment*)

Processo CRISP-DM

O CRISP-DM, como ilustra a imagem a seguir, é um processo cíclico e incremental que não termina com a implantação (*deployment*) do sistema. Ao contrário, o círculo externo simboliza a natureza cíclica do processo, incrementando novas funcionalidades com as lições aprendidas em cada ciclo.



Processo CRISP-DM.

Neste módulo, abordaremos as atividades de KDD que compõem as etapas de entendimento do negócio (*Business understanding*) e dos dados (*Data understanding*), para caracterização do problema a ser atacado no projeto de sistema de aprendizado de máquina.

As etapas de entendimento do negócio

Entendendo o problema

O entendimento do problema é fundamental para iniciarmos qualquer projeto de aprendizado de máquina. O entendimento do problema envolve as seguintes etapas:

Levantamento de requisitos

Segundo o processo do *Cross Industry Standard Process for Data Mining* (CRISP-DM), o profissional deve se reunir com os *stakeholders* do projeto para primeiro entender a demanda a que o projeto atenderá. Para isso, podemos aplicar técnicas clássicas de levantamento de requisitos, ou podemos beber nas fontes de metodologia ágil, como o Design Thinking, para mapear a dor dos clientes.

Reunião com especialistas

Após o levantamento do problema, ouviremos os especialistas no domínio (se houver na organização) e exploraremos os processos de negócio da organização (se houver).

Elaboração do documento para validação

Ao fim da compreensão do negócio e do problema, transcreveremos isso em um documento que será validado pelos *stakeholders*.

Uma vez que entendemos bem o problema, e após sua validação pelos *stakeholders,* partimos para o levantamento bibliográfico, pois precisamos saber se a literatura já provê soluções ou métodos para resolver tal problema. O levantamento bibliográfico, dependendo da organização, pode requerer um refinamento mais rigoroso e um processo metodológico mais robusto (no contexto acadêmico, temos as revisões sistemáticas de literatura, o mapeamento sistemático do estado da arte etc.), mas, para o nosso contexto didático e principalmente o da maioria das organizações, podemos simplificar para uma busca convencional.

Primeiro, pense em palavras-chaves que caracterizem o problema levantado, pois elas constituirão a primeira cláusula da nossa string de busca; depois, vamos analisar o problema e tentar encaixá-lo em uma das categorias de tarefas de aprendizagem de máquina respondendo a questões como:

1. O problema é de classificação ou clusterização?

2. Pode ser revolvido via aprendizado supervisionado, não supervisionado, semissupervisionado ou por reforço?

Dado que conseguimos encaixar o problema em uma dessas categorias, agora pensaremos numa possível solução:

1. Um modelo preditivo resolve?

2. Ou um modelo diagnóstico seria mais adequado?

3. Ou, ainda, seria melhor um modelo exploratório?

E se, por acaso, você não tiver certeza, não tem problema! É para isso que serve a atividade de levantamento bibliográfico. Caso seja experiente e queira buscar soluções que usem um tipo específico de algoritmo, também pode incluí-lo.

Agora temos 3 eixos com palavras-chaves: o eixo do problema, o eixo da tarefa de aprendizagem de máquina e, finalmente, o eixo de solução (que é facultativo). Cada um desses eixos se transformará em uma cláusula, e cada cláusula terá um conjunto de termos (as palavras-chaves). Para converter isso em uma string de busca, faremos da seguinte maneira:

```
(<PPC1> OR <PPC2> OR ) AND (<TPC1> OR <TPC2> OR <TPCN>) AND
(<SPC1> OR <SPC2> OR <SPCN>)
```

Essa string de busca significa que o motor de busca escolhido procurará artigos, links, trabalhos e soluções que abordem o contexto do projeto definido pelas cláusulas conjuntivas (separadas por AND) e que contenham pelo menos um dos termos de cada eixo (palavras-chaves separadas por OR).

Por exemplo:

```
(Classificação) AND (Finanças) AND (Árvores de Decisão)
```

Essa string procurará trabalhos relacionados que utilizem Árvores de Decisão no Âmbito de Finanças.

Observe a próxima string:

```
(Aprendizado Supervisionado) AND (Bolsa de Valores OR Ações OR
Financial Forecasting) AND (Modelos Preditivos)
```

Essa string procurará trabalhos relacionados de aprendizado supervisionado que utilizem algum modelo preditivo para fazer análise de ações (*Stock Analysis*).

Dica

É recomendado que a string seja sempre convertida para inglês, pois o alcance é muito maior.

Tendo a string de busca pronta, agora escolhemos um motor de busca para levantarmos o que está sendo feito na comunidade para resolver o nosso problema, ou um problema similar. Não existe um consenso de qual seria o melhor motor de buscas ou qual a melhor biblioteca digital para extrair esses trabalhos relacionados, mas utilizaremos os mais comuns na área de computação, que são a IEEE(*Institute of Electrical and Electronics Engineers*), a ACM(*Association for Computing Machinery*), a Elsevier, ou, caso não tenha acesso a essas, pode-se utilizar o Google Scholar.

Escolhida(s) a(s) fonte(s) de pesquisa, aplicamos nossa string de busca e analisamos o retorno parcial; se por acaso você estiver vendo muitos trabalhos não relacionados ou não estiver vendo quase nada (a busca retornou poucos trabalhos), é necessário refinar a string de busca. Normalmente resolvemos isso adicionando mais sinônimos de termos, ou reduzindo o número de cláusulas conjuntivas (caso tenham vindo poucos trabalhos), ou adicionando mais cláusulas conjuntivas (caso tenham vindo muitos trabalhos).

Dica

O resultado satisfatório não tem um gabarito, mas em geral a quantidade aceitável seria em torno de 100 trabalhos inicialmente, pois ao fim vamos analisá-los mais a fundo e eliminar por volta de 60% a 70% através da priorização de critérios de relevância ao problema.

Feito o levantamento dos trabalhos relacionados, procuraremos os seguintes itens:

1. Bases de dados utilizadas para treinamento de modelos.

2. Características mais relevantes para a solução.

3. Algoritmos mais utilizados para resolver o problema.

4. Métricas mais utilizadas para o monitoramento do modelo.

5. Melhores métricas (maior acurácia alcançada por exemplo).

Com esses 5 itens, podemos montar um plano de ação, ou plano do projeto, em que estarão descritos o problema, a tarefa de aprendizagem a ser utilizada, o modelo que será implementado, a métrica que será utilizada para acompanhar e o valor de base para comparação, bem como um esboço do processo de resolução do problema (pode ser uma extensão do CRISP-DM ou até mesmo do KDD). Esse documento será revisitado durante todo o processo para acompanhamento ou para atualização.

A coleta de dados

Caminho da coleta de dados

A coleta de dados é o processo de captura e medição de informações e variáveis de interesse, de forma sistemática, que permite responder a perguntas de pesquisa, bem como testar hipóteses e avaliar resultados. Existe uma diferença entre os tipos de dados qualitativos e quantitativos.

Dados qualitativos		Dados quantitativos	
Em sua maioria, são dados não numéricos, normalmente descritivos ou nominais. Costumam ser textos, sentenças ou rótulos. As perguntas que geram esse tipo de dados são abertas, e os métodos envolvidos são, frequentemente, grupos de foco, grupos de discussão e entrevistas.	São um bom jeito de mapear o funcionamento de um sistema ou a razão de um fenômeno. Mas, em contrapartida, são abordagens custosas que consomem muito tempo, e os resultados ficam restritos ao(s) grupo(s) de foco(s) envolvido(s).	São os dados numéricos, que podem ser matematicamente computados. Esse tipo de dado mede diferentes escalas que podem ser nominais, ordinais, intervalares e proporcionais. Na maioria dos casos, esse tipo de dado é resultado da medição de algum aspecto ou fenômeno.	Em geral, existe uma abordagem sistemática muito bem definida e mais barata de se implementar do que a coleta de dados qualitativa, uma vez que é possível construir processos automáticos ou simplesmente consumir relatórios gerados. Entre os métodos dessa categoria, temos os *surveys*, as *queries*, o consumo de relatórios e os escavadores (*scrapers*).

Esses tipos não são mutuamente exclusivos, pois é muito comum encontrar relatórios ou fazer entrevistas cujo resultado contenha tanto dados quantitativos (renda, número de familiares, despesas etc.) quanto dados qualitativos (endereço, sobrenome, instituição de ensino, grau, idade etc.).

No que tange à sua obtenção, os dados podem ser:

Dados primários

São aqueles coletados de primeira mão. Ou seja, dados que ainda não foram publicados, autênticos ou inéditos. Como é um dado recém-coletado, não tem interferência humana e, por isso, é considerado mais puro do que os dados secundários. Entre as fontes de dados primários, temos *surveys*, experimentos, questionários e entrevistas.

Os dados primários têm a vantagem de serem puros, coletados para a resolução de um problema específico e, se necessário, podem ser coletados de novo, a qualquer momento, para aumentar a quantidade.

Dados secundários

São aqueles dados que já foram publicados de alguma forma, ou seja, sofreram interferência humana. Por exemplo, ao fazermos a revisão de literatura em qualquer estudo, estamos revisando dados secundários. Entre as fontes de dados secundários, temos livros, registros, biografias, jornais, censos, arquivos de dados etc.

Os dados secundários são muito úteis quando não conseguimos fazer a coleta em primeira mão. Por exemplo, ao reproduzirmos um estudo de um trabalho da literatura, temos que utilizar os dados providos pelos autores. Entre as vantagens dos dados secundários, temos a economia de tempo em não ter que desenvolver um sistema de coleta, o menor custo e a delegação de responsabilidade (em relação aos dados) do profissional para o dono dos dados originais.

Métodos de coleta de dados primários

São métodos quantitativos ou qualitativos por meio dos quais você mesmo coleta os dados. O diferencial para esse tipo de método é a autenticidade dos dados, bem como a pureza deles, uma vez que você é a primeira pessoa a trabalhar com eles e, até o ato de publicação, apenas você tem acesso a eles.

Como vimos, a coleta de dados pode ser feita das mais variadas formas. A seguir, vamos conhecer esses métodos com mais detalhes.

Método de questionário

É um instrumento de pesquisa constituído de uma série de perguntas e interfaces para coleta de respostas e informações dos respondentes. Diferentemente das entrevistas ou dos levantamentos (*surveys*) tradicionais, o questionário é delimitado por suas opções de resposta, que estão alinhadas ao objetivo do projeto em questão. Alguns questionários contêm perguntas sobre variáveis individuais, no entanto, é possível observarmos questionários que lidem com agregação de valores com escalas, médias etc.

Método de entrevistas

É uma abordagem parecida com o questionário, diferenciando-se no meio de comunicação com os participantes, uma vez que a entrevista pode ser cara a cara, individual ou em grupo. Também pode ser feita por telefone ou meios eletrônicos, além de computadores. A entrevista em si pode ser estruturada, semiestruturada ou desestruturada, dependendo do objetivo do entrevistador. Basicamente, o método de entrevista é mais utilizado na coleta de dados qualitativos, principalmente para levantar processos, métodos etc., através de perguntas abertas na maioria dos casos.

Discussões em grupos de foco

É um método que junta um grupo pequeno e homogêneo de pessoas para discutir em profundidade sobre uma agenda de estudos. Basicamente, é uma entrevista em grupo com mais profundidade e com um foco maior, como diz o nome. A vantagem dessa dinâmica está na colaboração entre os participantes, estimulando-os a compartilharem mais informações e, consequentemente, mais dados para o projeto.

Levantamento (*Survey*)

É um método quantitativo feito por meio de questionários estruturados, podendo ser definido como uma maneira de coletar dados e informações a partir de características e opiniões de grupos de indivíduos. O resultado desse método, se feito em um grupo representativo, pode ser generalizado para uma população.

Estudo de caso

É um método de coleta de dados ou de experimentação capaz de extrair dados levando em conta o contexto observado. Tem como objetivo produzir conhecimento a respeito de um fenômeno observado, de modo a simulá-lo a partir desses dados contextualizados.

Amostragem de atividade

É um método de extração no qual diversas observações são feitas em série, durante períodos definidos, sobre um computador ou grupos de computadores, processos ou sistemas. É um dos métodos mais tradicionais usados no contexto de computação.

Métodos de coleta de dados secundários

Se os métodos de coleta primária demandam um esforço de aquisição maior em prol da especificidade e do maior controle dos dados, o método de coleta secundária inverte essa perspectiva. Basicamente, o profissional coleta os dados de repositório de dados, artigos ou patentes. Os exemplos mais comuns de coleta de dados secundários são repositórios públicos on-line como:

Kaggle

Website que tem como foco a viabilização de competições de Ciência de Dados on-line.

UCI

Repositório de conjunto de dados da Universidade da California em Irvine, que tem a vantagem de agrupar os conjuntos por tarefa de aprendizado, páginas públicas como a do IBGE etc.

> Qualquer que seja o repositório, é necessário sempre se ater à sistematização do processo, isto é, não podemos entrar nesses repositórios e coletar tudo que eles têm.

Quando aplicável, como o Kaggle ou o UCI, é recomendável utilizar o seu motor de buscas para rodar a string de busca que definimos na fase de entendimento do problema no processo CRISP-DM.
Outra alternativa são os artigos publicados nas bibliotecas digitais ou, ainda, as patentes que podem dar alguma dica de onde coletaram os dados de treinamento de seus modelos. Tal informação costuma ser encontrada nas seções 2 ou 3 dos artigos que tratam de experimentos ou estudos de caso, uma vez que essas seções descrevem a fundamentação teórica ou a descrição de dados primários ou secundários do artigo.

No caso de patentes, essa informação, se disponibilizada, costuma estar no corpo da descrição da patente. Para o levantamento de patentes ou artigos, podemos utilizar as ferramentas do Google, por facilitarem o processo, como o Google Scholar para artigos e o Google Patents para patentes.

Visão geral das etapas de entendimento do negócio e dos dados

O especialista Fernando Cardoso Durier da Silva fala resumidamente sobre os tópicos abordados neste módulo.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

Questão 1

A quem recorremos para validar o problema tratado por um projeto de aprendizagem de máquina?

A

Aos *stakeholders* do projeto.

B

Aos trabalhos relacionados.

C

Aos funcionários do departamento jurídico.

D

Aos funcionários do departamento de recursos humanos.

E

Ao gerente de projeto.

A alternativa A está correta.

Os *stakeholders* do projeto são os únicos atores capazes de explicitar qual o problema que eles estejam passando.

A opção b) não se aplica porque os trabalhos relacionados originam do problema.

As opções c) e d) só seriam válidas se os funcionários citados fossem os *stakeholders* do projeto, mas nem sempre eles estão envolvidos com os projetos da empresa.

A opção e) não é totalmente verdade, pois nem sempre o gerente de projetos está ciente dos problemas dos *stakeholders* e, assim, ofereceria uma visão unilateral insuficiente.

Questão 2

Um levantamento (*survey*) é implementado através de qual outro método de coleta adaptado?

A

Questionário quantitativo.

B

Questionário qualitativo.

C

Entrevista em grupos.

D

Grupos de discussão de foco.

E

Dinâmica de grupos.

A alternativa A está correta.

O questionário quantitativo é o método de coleta que, adaptado para o levantamento de dados quantitativos, implementa o *survey*.

A opção b) não está correta porque aborda o questionário qualitativo, cujas perguntas são abertas em sua maioria e não atendem à necessidade de especificidade que o levantamento precisa.

A opção c) é um método qualitativo e não atende à necessidade de especificidade que o levantamento precisa.

A opção d) é um método que fortalece o método de grupos de discussão, que, por sua vez, são para coleta qualitativa de dados por meio de perguntas abertas.

A opção e) não é um método de coleta de dados.

A etapa de preparação dos dados

Neste módulo, abordaremos as atividades de KDD que compõem a etapa de preparação dos dados (*Data preparation*) do processo CRISP-DM, em que ocorre a limpeza e o pré-processamento dos dados. Uma vez entendido o problema do negócio e seus dados, é realizada a sua preparação e os dados são validados para a escolha dos modelos adequados de aprendizado de máquina a serem treinados.

Conhecendo os dados

Agora que estamos na posse dos nossos dados, precisamos conhecê-los para conferir se eles estão de acordo com nosso plano de ação ou se precisam ser trabalhados ainda de alguma forma. Para isso, existem várias técnicas, como:

Extração de estatísticas descritivas

A extração de estatísticas descritivas dos atributos que compõem a base de dados, estatísticas como média, variância e desvio padrão. A partir dessas estatísticas, podemos checar qual é a característica dos nossos dados, ou seja, se eles são consistentes, regulares em torno da média ou se têm muita variação em relação à média.

Análise da distribuição dos dados

A análise da distribuição dos dados é outra estratégia bastante utilizada, muito importante para sermos capazes de fazer correlações entre os dados. Por exemplo, para dados numéricos, se a distribuição deles se aproximar da distribuição normal (ou o sino da curva gaussiana normal), isso indica normalidade dos dados, requisito importante para certos tipos de testes estatísticos.

As distribuições podem conferir características operacionais aos dados também, por exemplo, se observamos uma distribuição que se aproxima de uma distribuição de Poisson, é um forte indicativo de que estamos lidando com séries temporais, que por si só pertencem a uma categoria muito específica de problemas de aprendizado de máquina que, por sua vez, tem processos muito bem definidos a serem seguidos.

O estudo da distribuição dos dados pode ser feito a olho nu, mas isso não é o indicado, pois podemos nos enganar; o correto é extrair estatísticas como Skewness e Kurtosis, que, dependendo do valor, indicam que a distribuição é "achatada" ou é "pontuda", ou tem cauda pesada à esquerda ou à direita.

Testes estatísticos paramétricos

E, finalmente, temos testes estatísticos paramétricos, como o teste de Shapiro-Wilk, que, de fato, indicam a normalidade ou não da amostra; isso é feito por meio da observação do P-Value do teste, que se for menor do que o intervalo de confiança (normalmente 0.05, ou seja, 95%), dizemos que a amostra é diferente da Normal. Caso contrário, aceitamos a hipótese nula de que a amostra testada vem da mesma distribuição da amostra de referência (no caso do teste de Shapiro-Wilk, a Normal).

Valores nulos e dados repetidos

Um problema muito comum na atividade de pré-processamento de dados é a qualidade dos dados. Os dados podem vir certinhos, consistentes, talvez muito variados, mas, pelo menos, completos. Entretanto, é possível

que os dados venham com atributos faltantes, registros nulos, registros mal escritos (desformatados) etc. Para esses casos, existem diversas maneiras de resolver o problema, dependendo do tamanho da base, do tipo de processo de extração e da importância do atributo prejudicado.

1

Dados faltantes ou nulos em bases grandes

Dados faltantes ou nulos em bases grandes (por volta da ordem de grandeza de 10.000 registros ou mais) podem ser resolvidos ignorando o registro todo ou removendo-o da base, se a proporção de nulos não for expressiva (não passar de 10% da quantidade de registros). Essa estratégia é comum para bases grandes, pois a remoção desses registros nulos não será tão danosa ao processo de treinamento. Outra estratégia para esse caso específico é utilizar técnicas de regressão para dados numéricos ou de classificação para dados categóricos, para o preenchimento automático desses dados. O fato de a base ser grande ajuda o algoritmo de preenchimento automático, sendo claro que dados com variância alta podem prejudicar esse processo.

2

Dados faltantes ou nulos em bases restritas

Para dados faltantes ou nulos em bases de dados muito restritas ou pequenas (por volta da ordem de grandeza de 1.000 ou menos), temos duas alternativas: ou tentamos preencher de forma automática, como vimos, ou voltamos ao processo de coleta e tentamos melhorá-lo a fim de consertar o problema que causou a nulidade ou falta.

3

Dados faltantes ou nulos em bases de dados precárias

Para dados faltantes ou nulos em bases de dados precárias onde a exclusão do registro ou a interpolação dele seja inviável, o correto é retomar diretamente ao processo de coleta, pois claramente os dados serão insuficientes para o projeto.

4

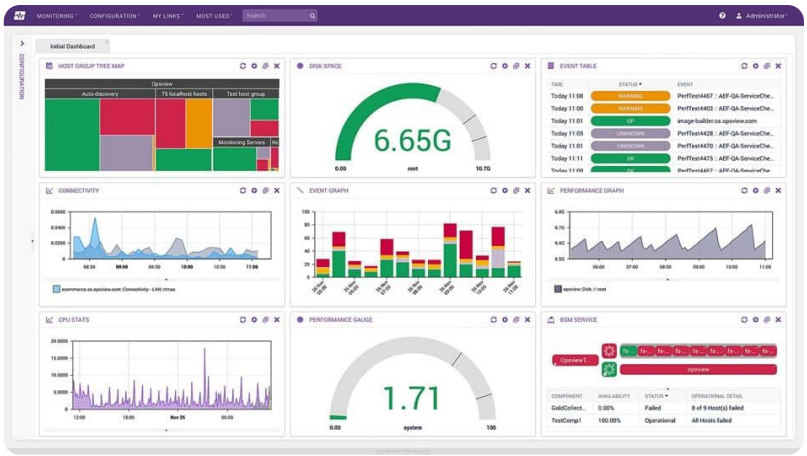
Dados repetidos

Para dados repetidos, a solução costuma ser simples! Basta eliminar os dados repetidos, a não ser que a repetição seja apenas para um subconjunto de atributos. Se for o caso, provavelmente a "repetição" ou é uma decomposição de uma agregação, ou é uma repetição de um evento em diferentes períodos de tempo. Para saber do que se trata, é sempre útil estudar os metadados do conjunto de dados, se estiverem disponíveis, ou estudar a origem deles.

Análise exploratória

A análise exploratória é uma pré-análise para a preparação dos dados e o treinamento do modelo. Aqui, combinaremos todo o aprendizado das subseções anteriores de conhecimento do dado, de resolução de dados nulos, e começaremos a experimentar os dados.

Na análise exploratória, começamos a tentar enxergar nos dados a concretização de nossas teorias sobre a resolução do problema e fazemos isso por meio da análise de correlação dos dados entre si. Podemos fazer a correlação entre cada atributo e o atributo alvo, procurando saber como os dados se distribuem geometricamente no espaço amostral (factível através de algoritmos de redução de dimensionalidade). Com isso, podemos conferir a importância de certas características, quais características contribuem mais ou menos para a classificação ou regressão, qual tipo de modelo utilizaremos – linear ou não linear.

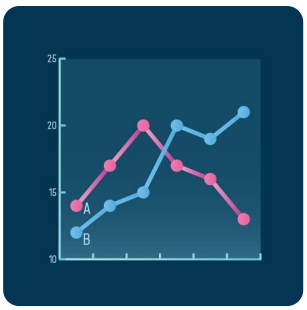


Dashboards de análise exploratória.

Na atividade de análise exploratória, podemos depender de ferramentas de *analytics* como Tableau, Power BI, Google Analytics ou Kibana para facilitar a visualização dos dados e explorá-los nos *dashboards*. Nestes, podemos criar:

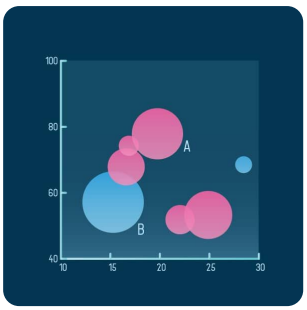
Gráficos de linha

Para entender a temporalidade dos dados ou a distribuição deles em sequência.



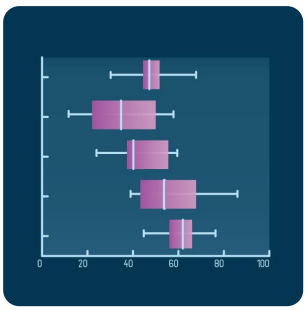
Gráficos de dispersão (scatterplots)

Para entender a correlação entre variáveis numéricas ou saber sua distribuição geométrica no espaço amostral.

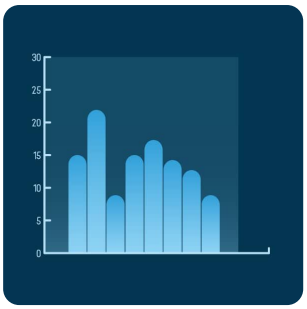


Gráficos boxplot

Para entender a variância dos dados ou, ainda, comparar estratos de uma característica.



Histogramas
Para variáveis numéricas.

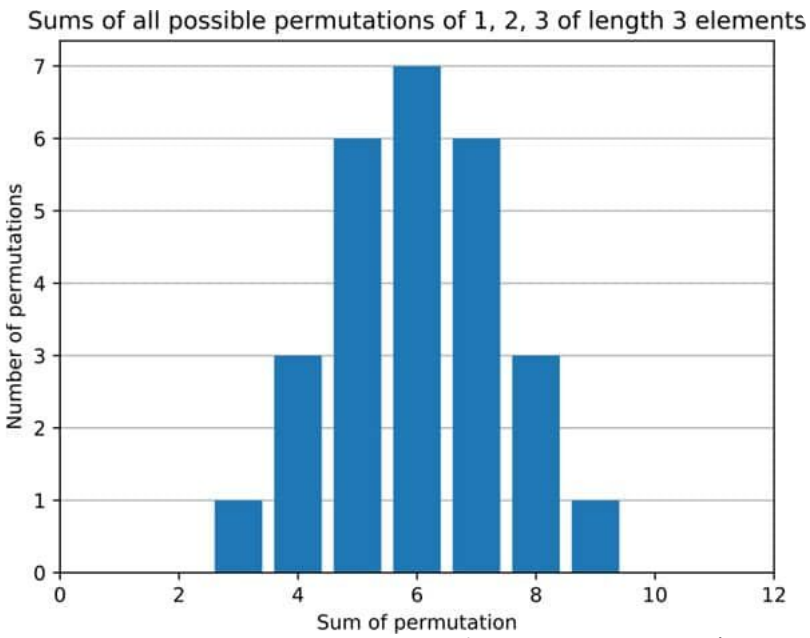


Gráficos de barra
Para variáveis categóricas, com o intuito de descobrir a distribuição dos dados e elucidar dados *outliers* (fora da curva).

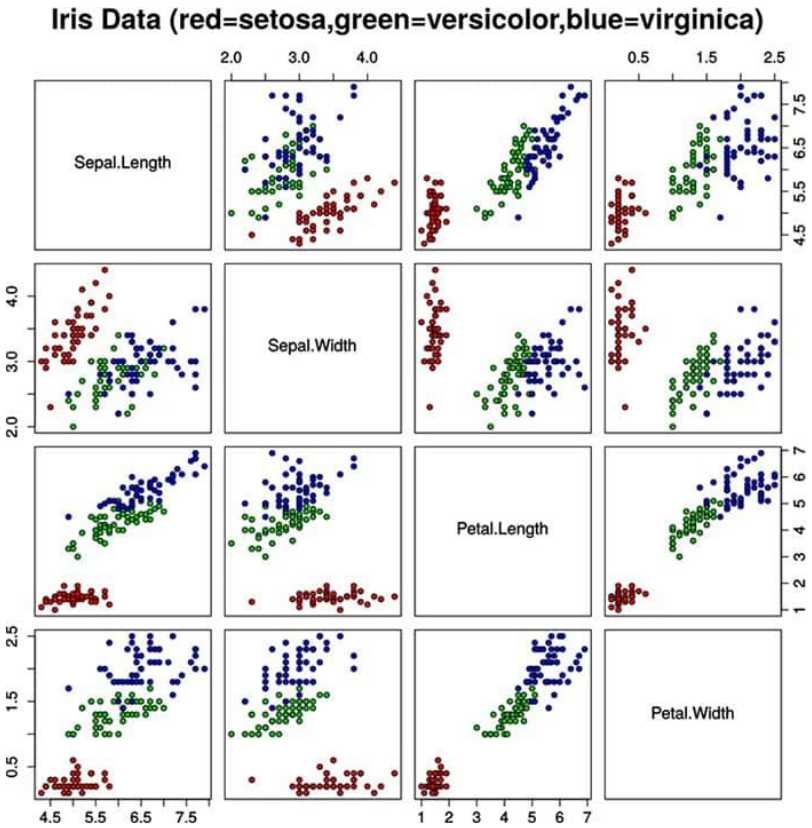
Dica

Podemos incorporar os gráficos no mesmo código em Python, por exemplo, com o uso do Plotly, Seaborn ou Matplotlib.

Operacionalizando a análise exploratória, podemos criar pipelines de extração de estatísticas descritivas, testes de correlação de Pearson ou Spearmen, plotagem de gráficos, quando utilizamos a estratégia de fazê-los via linguagem de programação (Python ou R), como vemos nas imagens a seguir.



Histograma mostrando a distribuição de valores (segue uma normal com média em 6)



Scatterplot de atributos do conjunto de dados Iris.

Saiba mais

O conjunto de dados Iris sobre flores ou conjunto de dados Iris de Fisher é um conjunto de dados multivariados introduzido pelo estatístico e biólogo britânico Ronald Fisher (1890-1962) como um exemplo de análise discriminante linear.

Para efeito de análise nos *dashboards*, é recomendado que coloquemos no eixo Y um outro atributo numérico do conjunto de dados ou o resultado de uma agregação. Para o eixo X, escolhemos dados categóricos para serem contabilizados ou datas/tempo ou outros atributos numéricos ou o atributo alvo. E, dependendo da análise, podemos colorir o gráfico de acordo com a distribuição de um dos eixos ou do atributo alvo, o que é muito útil na análise do *scatterplot* para verificarmos a linearidade ou não do nosso problema.

Preparação dos dados para a fase de treinamento

Visto que fizemos toda a análise exploratória, removemos os nulos, tiramos as repetições desnecessárias e entendemos os nossos dados, vamos empacotá-los para treinar o modelo de aprendizagem de máquina. Para isso, primeiro entenderemos quais modelos podem aceitar nossos dados, caso um modelo já não tenha sido escolhido, ou adaptaremos nossos dados ao modelo escolhido.

Um exemplo clássico de preparação de dados para a fase de treinamento, além de toda a limpeza feita, são os **sistemas de recomendação ou modelo de regras de associação**. Esse tipo de modelo específico exige que conjuntos de itens ou compras sejam agregados (como cestas de mercado); tais conjuntos são agregados por

um identificador e rotulados com o atributo alvo respectivo e, assim, passados adiante para o modelo, que emitirá recomendações no sentido de quem levou os itens x e y nas condições w, também levou o item z.

> Redes neurais precisam que os dados não só sejam numéricos como também amostrados em bateladas (*batch*) de tamanho definido, para serem treinadas iterativamente através das épocas.

No processamento de linguagem natural, temos a vetorização de textos, ou construção de saco de palavras (*bag of words*), que é a transformação dos dados textuais em matrizes esparsas de correspondência de palavras (colunas) em documentos/registros (linhas).

Cada solução e cada modelo demandará um tipo de preparação dos dados para a fase de treinamento. Outra prática interessante para otimizar o processo de treinamento de modelos é a remoção de *outliers* por meio da análise do intervalo interquartil dos atributos analisados, para eliminar os registros cujos atributos relevantes ultrapassem 0.5 do intervalo interquartil superior e inferior.

Outra estratégia para otimizar o processo de treinamento, bem como não viciar o modelo em certas escalas numéricas, é fazer a normalização do conjunto de dados, ou seja, para cada atributo, aplicar a normalização de escala normalmente dada pela estratégia MinMax, que recalcula os valores do atributo enquadrando-os na proporção da faixa de valores entre o máximo e o mínimo possíveis do atributo. Existe a variação na qual é possível definir os extremos da escala e fazer o cálculo de proporção nesse intervalo, por exemplo -1, +1 para distribuições positivas, neutras e negativas.

Demonstração em Python

Para implementar o estudo, o pré-processamento e a preparação dos nossos dados, precisaremos instalar as bibliotecas a seguir.

```
python

import pandas as pd
import numpy as np

import plotly.express as px
import seaborn as sb
import matplotlib.pyplot as plt

from sklearn.datasets import load_iris
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
```

Feito isso, configuraremos o *dataset* usando a base de dados Iris. Primeiro, carregaremos o *dataset*, depois criaremos a função label, que irá rotular nosso conjunto com o nome das classes, em vez do código numérico da classe.

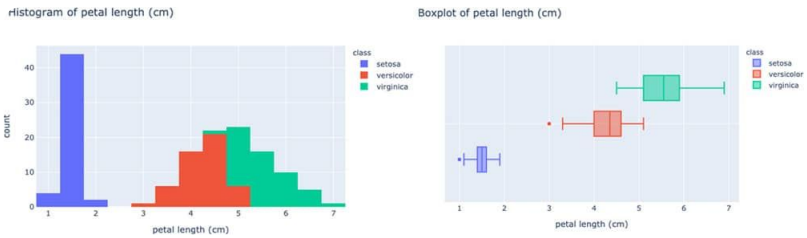
Para aplicar ao *dataset*, usaremos a função lambda do pandas. A função rótulo basicamente mapeia a classe para o código numérico através do index do vetor de nomes.

```
python

iris = load_iris()
def label(x):
labels = iris.target_names
return labels[x]
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
iris_df['class_codes'] = iris.target
iris_df['class'] = iris_df['class_codes'].apply(lambda x: label(x))
```

Agora, vamos entender o comportamento dos dados. Faremos isso imprimindo o *boxplot* e o histograma de cada coluna, comparada com a classe. O *boxplot* nos mostra os intervalos interquartis, o que pode ser útil para detectar *outliers*, enquanto o histograma apresenta a distribuição dos dados, bem como as escalas.

```
python

for c in iris_df.columns:
boxfig=px.box(iris_df, x=c, color='class', title='Boxplot of '+c)
boxfig.show()
histfig=px.histogram(iris_df, x=c, color='class', title='Histogram of '+c)
histfig.show()
```

Tal exemplo gerará gráficos como os ilustrados a seguir, só que para cada atributo do *dataset*.



Exemplo de histograma e boxplot. Gerado pela demonstração em Python.

Observando os dados, podemos perceber que eles não apresentam *outliers* significativos, porém os atributos estão em escalas diferentes; ainda que estejam todos em centímetros, podemos ver que comprimento de pétala e largura de sépala, por exemplo, estão em escalas diferentes em ordem de grandeza. Para regularizar isso, vamos utilizar o MinMaxScaler.

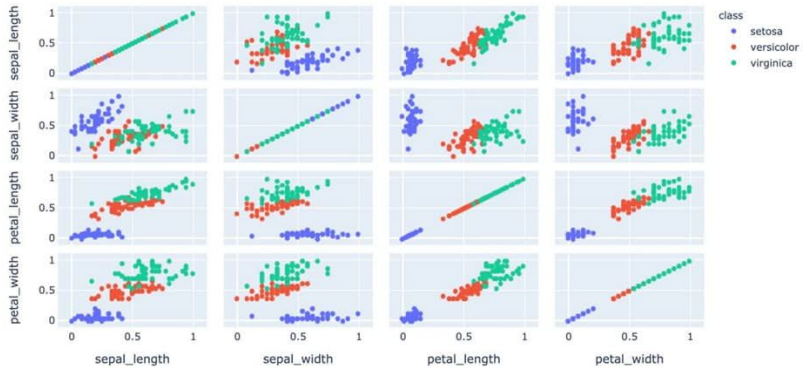
```
python

num_columns = iris_df.select_dtypes(include=numerics)
to_standardize = num_columns.columns.tolist()
to_standardize.remove('class_codes')
scaler = MinMaxScaler((0,1))
iris_df[to_standardize] = scaler.fit_transform(iris_df[to_standardize])
```

Nem todas as colunas devem ser regularizadas pelo MinMax para não descaracterizar o *dataset*, apenas os atributos numéricos. A classe codificada ou qualquer atributo que seja numérico, mas categórico por natureza (notas, *ranks*, idade, código etc.), não deve ser regularizado.

Agora que temos tudo regularizado, podemos continuar. Vamos visualizar a correlação entre os atributos do conjunto de dados. Para isso, faremos:

```
python

fig = px.scatter_matrix(iris_df, color='class')
fig.show()
```

Com o plotly, vamos gerar a matriz de *scatterplot*, que produzirá algo como na imagem a seguir.



Scatter Matrix. Gerado pela demonstração em Python

Feito isso, tentaremos reduzir a dimensão do *dataset*.

```
python

pca_comp=2
pca = PCA(n_components=pca_comp)
feature_columns = [ c for c in iris_df.columns if c not in ['class','class_codes'] ]
pca_iris_df = iris_df[feature_columns]
pca_iris_df = pd.DataFrame(pca.fit_transform(pca_iris_df), columns=['pc_'+str(i) for i in
range(0,pca_comp)])
pca_iris_df['class_codes'] = iris_df['class_codes']
```

Mais uma vez, para não prejudicar o rótulo do *dataset,* vamos retirá-lo da lista de colunas do *dataset*, para que o **PCA** possa ser treinado adequadamente. Feito isso, produziremos o novo *dataset* pca_iris_df, que tem dimensão reduzida e está regularizado com os dados na mesma escala (entre 0 e 1, como vimos no MinMaxScaler).

PCA

Análise de Componentes Principais – do inglês Principal Component Analysis: técnica de análise multivariada usada para descobrir interrelações entre variáveis e explicar essas variáveis em termos de suas dimensões ou componentes.

Demonstração da etapa de preparação de dados

O especialista Fernando Cardoso Durier da Silva demonstra as atividades de limpeza e pré-processamento na etapa de preparação de dados do CRISP-DM.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

Questão 1

Qual o benefício da análise exploratória para o projeto de aprendizado de máquina?

A

Compreensão do conjunto de dados.

B

Estudo da lógica do algoritmo de aprendizagem de máquina.

C

Melhoria dos processos de negócio da organização.

D

Mapeamento do estado da arte.

E

Estabelecimento do valor de base para comparação de métricas.

A alternativa A está correta.

A análise exploratória visa primordialmente ao entendimento do conjunto de dados, como pré-análise para a preparação dos dados e o treinamento do modelo.

A opção b), no contexto de projeto de aprendizado de máquina, só é obtida a partir do estudo do algoritmo em si ainda na fase de entendimento do problema no levantamento bibliográfico.

A opção c) é o objetivo final do projeto.

A opção d) é uma atividade diferente da análise exploratória.

A opção e) faz parte do levantamento bibliográfico da fase de entendimento do problema.

Questão 2

Ao regularizarmos os atributos do conjunto de dados com a estratégia MinMax, nós queremos:

A

Otimizar o tempo de treinamento.

B

Remover vícios de escala do modelo.

C

Aumentar o tamanho da base de dados.

D

Resolver o problema de dados faltantes.

E

Remover valores repetidos.

A alternativa B está correta.

O objetivo da regularização dos atributos do conjunto de dados com a estratégia MinMax é normalizar os valores num intervalo predeterminado, para otimizar o processo de treinamento, de modo a não viciar o modelo em certas escalas numéricas.

A opção a) não tem relação com a regularização MinMax.

A opção c) é feita por meio de melhoria do processo de coleta ou geração de dados sintéticos.

A opção d) é feita através de remoção de registros ou interpolação de dados.

A opção e) é resolvível por exclusão dos valores repetidos mediante análise prévia do conjunto.

A etapa de modelagem

Neste módulo, abordaremos as atividades de KDD que compõem a etapa de modelagem (*Modeling*) do processo CRISP-DM, em que ocorre a seleção do modelo e dos algoritmos de aprendizado de máquina, assim como o treinamento do modelo com dados de teste e de validação.

Criação dos conjuntos de treinamento, teste e validação

Com os dados limpos, pré-processados e prontos para serem usados, prepararemos nosso processo de aprendizado automático. Você pode estar se perguntando: Como faremos isso?

Da mesma maneira como fazemos quando educamos estudantes! Passamos conteúdo para ser estudado e absorvido por eles, apresentamos variados exemplos, aplicamos testes para observar como estão aprendendo e onde podem melhorar e, finalmente, aplicamos a avaliação final para ver se são capazes de generalizar soluções de problemas similares aos que estudaram. É assim que se baseia, também, o processo de aprendizado de máquina.

Dado que temos um conjunto, como vamos oferecê-lo ao algoritmo de aprendizado?

É bem simples! Começaremos dividindo-o em três conjuntos:

1

Conjunto de treinamento

O maior conjunto dos três utilizados pelo modelo para aprender, normalmente na proporção de 70% do conjunto original.

2

Conjunto de testes

É o conjunto "controlado" no qual o modelo vai aplicar seus conhecimentos e tentar classificar corretamente a variável alvo desse subconjunto; aqui, o modelo pode "errar" e ser reajustado, e, normalmente, a proporção desse subconjunto é de 15%, se houver conjunto de validação; ou 30%, se não tiver.

3

Conjunto de validação

Tem o objetivo de simular a realidade ou o "mundo real"; basicamente, até chegar aqui, o modelo aprendeu com o conjunto de treinamento, testou contra uma realidade controlada do teste e, agora, treinará com o "mundo real". Antes de ir para produção, o conjunto de validação tem a proporção de 15% do conjunto original.

Ainda na construção dos conjuntos de aprendizagem, temos a separação entre características e alvo para operacionalização do processo de aprendizagem. Com o conjunto de treinamento, apresentam-se as características e, em seguida, o alvo; depois, no teste e na validação, apresentam-se apenas as características na espera de que o modelo emita um parecer para ser avaliado.

Divisão clássica do conjunto de dados.

Como nem sempre nos deparamos com problemas de classificação binária, por exemplo, temos que considerar a estratificação dessa divisão para que os subconjuntos sejam amostras o mais semelhantes possível em relação ao conjunto original; caso contrário, o modelo poderá enviesar suas decisões para uma classe em detrimento da outra.

Isso também vale para regressões, pois, se a estratificação não for feita efetivamente, o modelo pode fazer com que a função de regressão fique enviesada para o estrato de maior valor. Para essa questão de desequilíbrio entre estratos, existe também a estratégia de **reamostragem de equalização**, que seria equalizar os estratos pelo menor, mas isso implica problemas como o distanciamento da realidade, além da perda de informação como efeito negativo.

Definição do algoritmo e configuração de arquitetura

A definição do algoritmo ou a escolha do modelo dependem de dois fatores no processo do CRISP-DM:

1. Definição do estado da arte de trabalhos relacionados.

2. Compatibilidade dos dados com o algoritmo candidato.

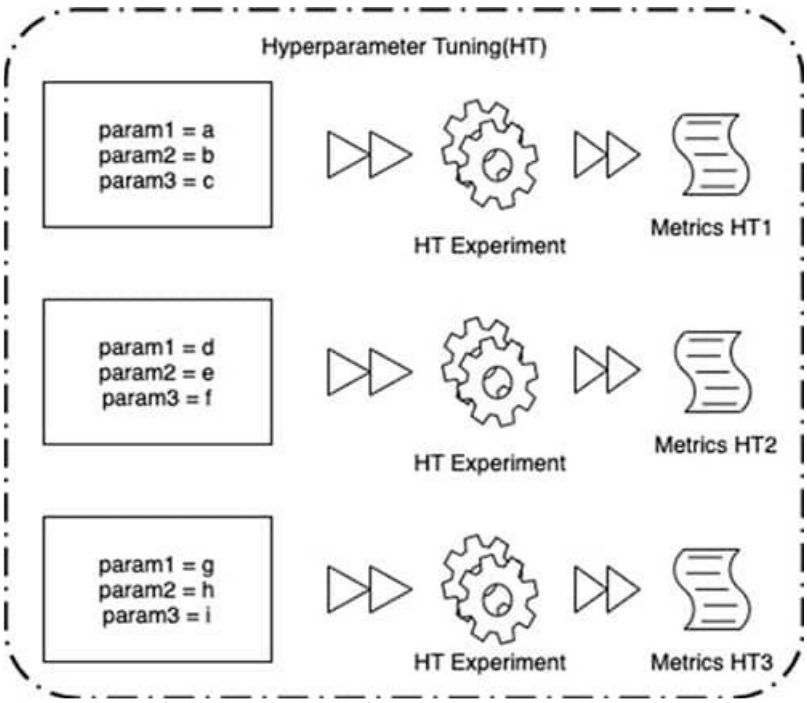
Feita a análise dos algoritmos recomendados com relação à compatibilidade com os dados, pode ser que ainda sobre mais de uma opção; no contexto de trabalhos mais acadêmicos ou experimentais, faz-se o treinamento com os todos os algoritmos e escolhe-se o que maximize a métrica de análise. No contexto industrial, costuma-se aplicar a regra da Navalha de Ockman, segundo a qual devemos escolher sempre o sistema/modelo mais simples entre aqueles que têm o resultado de suas métricas de comparação da mesma ordem de grandeza ou similares.

Exemplo

Se uma Rede Neural Convolucional tem 82% de acurácia e uma Floresta Aleatória tem 81% ou 80%, escolheríamos a Floresta Aleatória por ser menos complexa do que a rede neural em questão, de modo a ter um resultado de métrica muito próximo ao do outro modelo mais complexo.

A configuração de arquitetura do algoritmo está relacionada com dois aspectos: o primeiro é o aspecto do aprendizado, afinal, queremos otimizar o aprendizado automático, mas também tem o fator prático, ainda mais na indústria, onde muitas vezes temos situações nas quais o algoritmo tem que ser rápido e eficiente.

Para isso, nós fazemos a configuração de hiperparâmetros. Tal experimento empírico consiste em estabelecer uma tabela (ou estrutura similar, no caso prático em Python, GridSearch) onde as linhas serão as rodadas, ou experimentos, e as colunas serão os parâmetros do algoritmo.



Esquema de configuração de hiperparâmetros.

O experimento de configuração de hiperparâmetro consiste em rodar várias instâncias do algoritmo com diferentes configurações dos seus parâmetros, anotando os resultados de métricas de análise, conforme ilustrado no esquema de configuração de hiperparâmetros. Ao fim, estabelecemos os parâmetros por meio da escolha da configuração que maximiza as métricas. Isso também vale para arquiteturas de redes neurais, em que as variações são feitas no número de camadas escondidas, os tamanhos destas, bem como outros parâmetros não estruturais, como número de épocas, função de otimização etc.

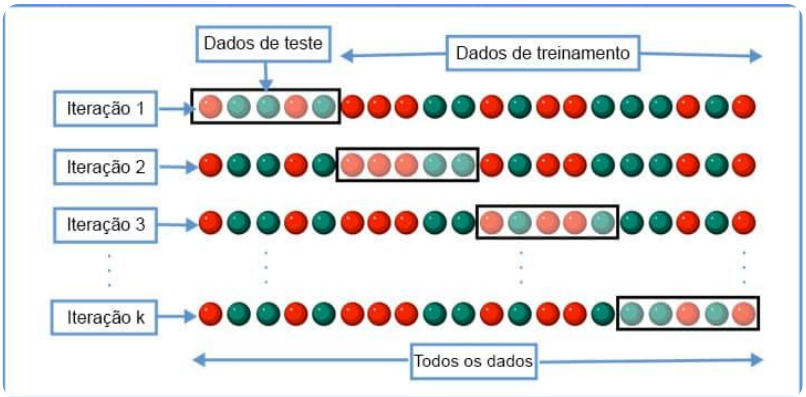
Análise do processo de treinamento e o aprendizado

O processo de treinamento deve ser acompanhado com cautela, uma vez que refletirá no sucesso ou na falha do projeto. Para tal, é necessário estabelecer um processo de experimentação bem definido.

> O modelo, na primeira atividade, será treinado com o conjunto de treinamento e será testado como o conjunto de testes.

Observamos os resultados da métrica de avaliação e, se o modelo tiver performado melhor do que 50%, então saberemos que ele não está reciprocando. Depois, analisamos se ele está performando melhor ou igual ao estado da arte; se sim, então ele está consistente ou melhor. Caso não tenha passado em alguma dessas condições, devemos voltar para a atividade de pré-processamento mais uma vez.

Agora, testaremos o modelo com o conjunto de validação e aplicaremos as mesmas condições de avaliação. Se tudo der certo, você terá um modelo pronto para ser implantado em produção. Já em produção, será necessário o monitoramento constante, visto que qualquer hesitação do modelo deve ser analisada e corrigida o mais rápido possível, pois pode significar consequências terríveis ao negócio!



Validação cruzada.

Mas, ainda assim, você pode se perguntar:

Reflexão

Será que esse é o melhor jeito? Será que a divisão do conjunto de dados não pode ser enviesada por seleção, mesmo que aleatória?

A resposta é sim! Isso pode acontecer, mas é um risco que corremos em troca de um processo menos complexo. Como pode ser visto na imagem anterior, a solução mais adequada para esse tipo de problema de viés de divisão é fazer a validação cruzada (*Cross Validation*), que consiste em iterativamente dividir o conjunto em k subconjuntos e circularmente treinar com k-1 partições, deixando uma partição de fora para ser avaliada, e fazer isso até que todas tenham sido essa partição de validação. A métrica de avaliação para esse processo é dada pela média das métricas de cada k testes. O efeito negativo desse tipo de processo é gerar um modelo super ajustado, caso o k seja muito grande.

Por fim, para validar tudo, é interessante extrair a função de decisão ou estrutura de decisão, caso seja possível, e avaliar com os *stakeholders* para ver se é compatível com a regra de negócio. Essa é uma das principais ocupações da área de explicabilidade de modelos, cujos maiores desafios estão relacionados aos modelos complexos, como os incorporados ou as redes neurais.

Avaliação do treinamento do modelo

Para avaliarmos se o modelo foi realmente treinado e se, de fato, é capaz de generalizar ou de atuar sozinho no contexto organizacional para o qual foi criado, temos que monitorar seu funcionamento em produção ou no ambiente de pré-produção, caso seja possível, para acompanharmos como ele lida com o dia a dia.

Outra opção, muito comum, principalmente com classificadores, é ter uma sessão de avaliação do modelo com especialistas. Basicamente, teremos um modelo funcionando, classificando várias observações, enquanto o(s) especialista(s) observa(m) e corrige(m) no meio do caminho. Ao fim desse arranjo, contabilizamos quantos acertos o modelo teve e quantas correções foram necessárias. Essa abordagem oferece como vantagem e desvantagem:

Vantagem

Garantia de que o modelo estará sempre atualizado e ajustado.

Desvantagem

Possível sobreajuste indesejado, o custo alto, pois seria necessário alocar os especialistas para acompanharem o modelo.

As métricas de avaliação obtidas no experimento antes da implantação são úteis para liberar uma versão viável para uso, mas não necessariamente refletem a realidade, uma vez que o conjunto de treinamento, por maior que seja, é apenas uma amostra do mundo real.

Atenção

É importante sempre monitorar, questionar e, principalmente, não ser intolerante a erros, pois é a partir deles que aprendemos a melhorar o processo experimental e os modelos. Por essa razão, os processos de mineração de dados ou descoberta de conhecimento, como o CRISP-DM e o próprio KDD, são cíclicos.

Demonstração em Python

Agora, seguiremos para o treinamento do modelo com o *dataset* Iris reduzido e regularizado. Para isso, precisaremos instalar mais bibliotecas.

```
python

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
```

Começaremos separando o conjunto de dados em treinamento e teste. Faremos isso utilizando a função do sklearn de separação de dados. A função calcula os tamanhos dos conjuntos, com base na proporção do tamanho desejado para o conjunto de testes.

```
python

feature_columns = [c for c in pca_iris_df.columns if c != 'class_codes']
X_train, X_test, y_train, y_test = train_test_split(
pca_iris_df[feature_columns],
pca_iris_df['class_codes'],
test_size=0.2,
random_state=0
)
```

Fizemos uma separação 80/20, ou seja, 20% para o conjunto de testes e 80% para o treinamento.

Dica

Caso seja necessário, podemos dividir o conjunto de testes de novo, para fazer o conjunto de validação, mas como o dataset só tem 150 registros para fim de aprendizado, é melhor não dividirmos mais (com datasets maiores, do dia a dia, é recomendado fazer um conjunto de treino, teste e validação).

Por último, treinaremos nossa floresta aleatória. Para isso, passaremos o X_train e o y_train para a floresta aleatória, enquanto, para testarmos o modelo, utilizaremos o X_test e o y_test.

```python
clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(X_train, y_train)
print(clf.score(X_test, y_test))
```

Demonstração do treinamento de modelos

O especialista Fernando Cardoso Durier da Silva demonstra o treinamento de modelos.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

Questão 1

Qual a proporção do tamanho do conjunto de testes comumente definida na literatura quando temos um conjunto de validação presente?

A

1%

B

15%

C

50%

D

60%

E

80%

A alternativa B está correta.

Ainda que não necessariamente exista um gabarito do tamanho da divisão, é importante termos um padrão (ou base de partida), que é adotado como 15% na prática.

Dito isso, a opção a) é muito ínfima para avaliar efetivamente.

A opção c) é exagerada, pois pode prejudicar o treinamento.

A opção d) é impraticável, já que o modelo provavelmente ficará desajustado.

A opção e) é inviável, seria como se aplicássemos uma prova de cálculo para um aluno do terceiro ano do fundamental.

Questão 2

Qual estratégia de treinamento divide o conjunto de dados em subconjuntos iterativamente e treina o modelo de forma circular, deixando um subconjunto para validação?

A

Divisão Treino, Teste e Validação.

B

Lista circular.

C

Espiral de treinamento.

D

CRISP-DM.

E

Validação cruzada.

A alternativa E está correta.

A validação cruzada consiste em iterativamente dividir o conjunto em k subconjuntos e circularmente treinar com k-1 partições, deixando uma partição de fora para ser avaliada, até que todas tenham sido essa partição de validação.

A opção a) só faz a divisão uma única vez.

A opção b) é uma estrutura de dados.

A opção c) não existe.

A opção d) é o processo utilizado para projetos de aprendizagem de máquina.

Avaliação de resultados e a implantação do sistema

Neste módulo, abordaremos as atividades de KDD que compõem as etapas de avaliação dos resultados (*Evaluation*) e implantação (*Deployment*) do processo CRISP-DM, em que ocorrem os testes do modelo de aprendizado de máquina selecionado, a avaliação dos resultados e a implantação do sistema no ciclo implementado. Assim, um *release* do sistema entra em produção, podendo ser incrementado com novas funcionalidades, à medida em que novas questões de análise forem surgindo com o uso do sistema.

Tipos de teste

Para a avaliação dos modelos de aprendizagem de máquina, existem muitas estratégias. Uma das mais comuns é o teste AB, ou seja, aplicar uma solução para um grupo/uma amostra e não aplicar nenhuma ou aplicar uma solução diferente para a outra amostra. A finalidade é medir o nível de eficácia das soluções, comparando os resultados obtidos. Esse é basicamente o princípio por trás da divisão tradicional de dados em treino, teste e validação.

Além disso, temos também os testes de hipótese, nos quais geramos distribuições e as comparamos a fim de saber se são diferentes ou iguais. Tal teste nos diz se houve melhora ou piora, e se o tratamento é o mesmo, mediante a análise do P-Value, isto é, a probabilidade de se obter uma estatística de teste igual ou mais extrema que aquela observada em uma amostra, sob a hipótese nula. Isso, no contexto de aprendizado de máquina, pode ser feito da seguinte maneira:

1. Instanciar o modelo para solução 1.

2. Instanciar o modelo para solução 2.

3. Estabelecer um laço experimental de 1000 rodadas (ou um número grande o suficiente para gerar um conjunto de observações estatisticamente significativas).

4. Dentro do laço experimental, repetir o processo de treinamento dos modelos por meio de validação cruzada e guardar os resultados de cada rodada, sendo que cada repetição de treinamento deve ser feita com uma amostra de 90% do conjunto original, amostrada aleatoriamente a cada passo.

5. Ao fim, executar teste de normalidade em cada uma das amostras resultantes.

6. Testada a normalidade, escolher um teste estatístico adequado e testar as distribuições (o teste de Wilcoxon é uma boa opção para distribuições não normais).

7. Caso o teste resulte em P-value menor do que 0.05, então, as amostras são diferentes, o que significa que um modelo é melhor do que o outro.

8. Para descobrir qual é o melhor modelo, basta analisar qual tem a maior média.

Saiba mais

Teste de WilcoxonÉ um teste de hipóteses não paramétrico utilizado quando se deseja comparar duas amostras relacionadas.

Os testes de hipótese não são muito comuns no meio industrial. Por outro lado, são mais comuns no meio acadêmico, quando queremos provar que nossa solução é melhor do que outra, seja porque nossa configuração de hiperparâmetros é diferente, seja porque fundimos dados ao conjunto de dados original etc. Mas tal processo metodológico na indústria pode ser usado para provar que uma solução é melhor do que alguma que já exista e seja ultrapassada, ou que é melhor do que nada (simulação dos processos correntes da organização).

Ainda existe o teste mais simples, que é a comparação de métricas de avaliação diretamente. Porém, o problema de testes feitos assim é que resultados muito próximos podem ser considerados diferenças estatisticamente insignificantes.

Exemplo

O resultado modelo1 🔢 84% e modelo2 🔢 85% levanta suspeita de ter havido um viés de rodada experimental, podendo, na próxima vez que for feito o experimento, resultar nos mesmos valores ou o inverso.

Métricas para avaliação, validação e teste

O melhor jeito de avaliar seu modelo de aprendizado de máquina é a matriz de confusão. Trata-se de uma tabela de contingência, na qual as linhas são os valores previstos e as colunas são os valores reais, e cada célula conta a quantidade de acertos na diagonal (quantas vezes a classe prevista era a real) e as demais, os erros de predição (foi prevista uma classe, quando na realidade era outra).

Quando a matriz de confusão é binária, os cálculos das métricas de avaliação são mais simples. Na matriz binária, são mais evidentes o erro do tipo 1 e o erro do tipo 2, que são, respectivamente, a predição de positivo, quando na realidade era negativo; e a predição de negativo, quando era positivo.

Da matriz de confusão, podemos extrair todas as métricas mais utilizadas da literatura, a saber:

Acurácia (accuracy)

Indica uma performance geral do modelo, ou seja, dentre todas as classificações, quantas o modelo classificou corretamente.

Precisão

Indica, dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas.

Revocação (recall)

Indica, dentre todas as situações de classe Positivo como valor esperado, quantas estão corretas.

Medida F (F-Measure)

Indica a média harmônica entre precisão e recall.

Veja um exemplo de matriz de confusão na tabela a seguir.

Acurácia	(VP+VN)/(VP+VN+FP+FN)	Detectada	
F-Measure	(2*Precisão*Recall)/(Precisão+Recall)	Sim	Não
Real	Sim	Verdadeiro Positivo (VP)	Falso Negativo(FN)
	Não	Falso Positivo(FP)	Verdadeiro Negativo(VN)
Precisão	(VP)/(VP+FP)	Recall	(VP)/(VP+FN)

Matriz de confusão e suas fórmulas.

Capacidade de generalização

O objetivo dos modelos de aprendizado de máquina é a capacidade não só de aprender sozinho e se adaptar, mas, principalmente, de generalizar, ser capaz de lidar com exemplos nunca vistos. Se o modelo só acertar os exemplos sobre os quais foi treinado, significa que está decorando, fazendo sobreajuste, o que é terrível.

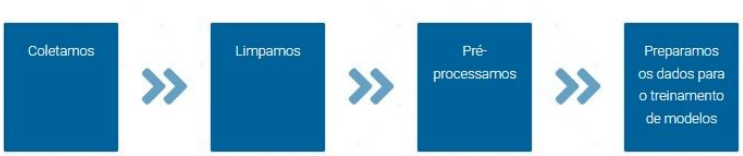
Para analisarmos a capacidade de generalização do modelo, devemos monitorar a medida F (F-Measure), pois, sendo uma média harmônica entre precisão e recall, saberemos se o modelo está acertando mais uma classe do que outra, mostrando que nosso modelo é consistente. Porém, para garantir que ele generalize, devemos desconfiar de valores acima de 90%, visto que seriam muito bons, embora possíveis. Depois, precisamos experimentar o modelo com dados reais, nunca vistos, e aferir a performance dele.

Atenção

Os modelos incorporados, bem como as redes neurais, se não tiverem seus hiperparâmetros configurados adequadamente, tendem ao overfit (sobreajuste) e, consequentemente, não serão capazes de generalizar.

Finalização do processo do CRISP-DM ou KDD

Até aqui, passamos por todas as fases do processo:



Agora, chegamos à etapa final do processo: a **implantação do sistema desenvolvido.**

> A implantação dos modelos de aprendizado de máquina no processo CRISP-DM consiste na incorporação do modelo proposto nos processos de negócio da organização.

Assim como qualquer software desenvolvido por profissionais da área, é necessário estabelecer métricas de monitoramento bem como políticas de *continuous integration* e *continuous delivery* (CI/CD). Porém, para os modelos de aprendizagem de máquina, existem testes específicos para permitir o **rollout**, como os testes de hipótese mencionados anteriormente.

Rollout

Rollout de TI consiste na migração de uma tecnologia, que pode ser instalada em alguns ambientes de uma organização ou até mesmo em todas as suas unidades de negócios.

Esse processo de implantação deve ser alinhado com a área de negócios para combinar por quanto tempo o profissional de ciência de dados prestará o suporte para aquele modelo e, depois do fim do período de suporte, fica a cargo da organização alocar outros profissionais para monitorar o modelo e reiniciar o processo de treinamento, codificado pelo criador do modelo.

Demonstração em Python

Vamos continuar nosso experimento do módulo 3. Agora, com a floresta aleatória instanciada, faremos o retreinamento dela utilizando a validação cruzada. Desta vez, porém, usaremos como parâmetro de métrica o 'f1_weighted', que é o indicador de que a validação cruzada utiliza a medida F para medir o modelo.

Dica

Recomendamos o uso da medida F porque é a média harmônica entre precisão e recall. Além disso, ela deve ser ponderada, pois o nosso problema é uma classificação multiclasse (mais de duas classes possíveis) e, também, pelo fato de considerar o desequilíbrio entre classes, que não é o caso do exemplo em questão, mas, por via de dúvidas, é sempre bom usar.

Existem outros indicadores, como o f1_micro e f1_macro, também necessários quando lidamos com classificação não binária. A diferença entre um e outro é que o macro leva em conta as predições por classe, enquanto o micro considera as predições globais.

```python
cv_result = cross_val_score(clf, pca_iris_df[feature_columns],
pca_iris_df['class_codes'], cv=5,scoring='f1_weighted')
print("CV Result: ", cv_result.round(3))
print("CV Mean: ", round(cv_result.mean(), 3))
print("CV Std: ", round(cv_result.std(),3))
```

Após essa etapa, mergulharemos a fundo nos resultados, começando por montar a matriz de confusão. Para isso, precisaremos dos valores verdadeiros, ou seja, os valores reais do conjunto de teste, e faremos uma simulação de predições das observações do conjunto de testes. Assim, temos o nosso *ground truth* (verdade fundamental) nos valores reais e os valores estimados gerados a partir das predições. Quando podemos dividir o *dataset* em treino, teste, e validação, o conjunto de validação deve ser usado aqui.

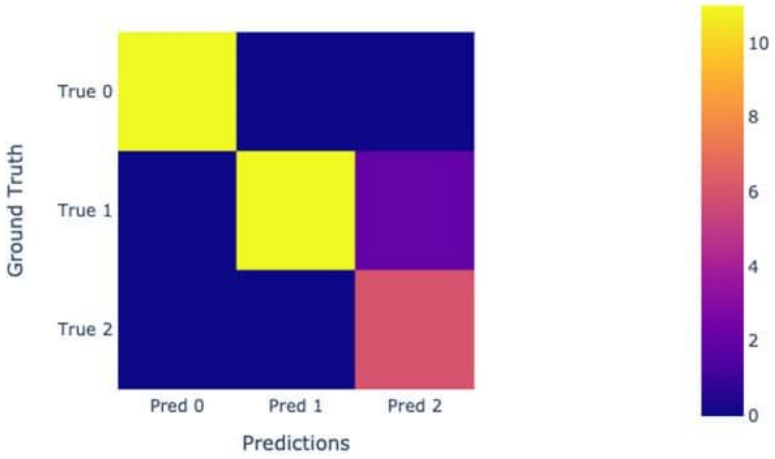
```python
y_true = y_test.values
y_pred = clf.predict(X_test)
cm = confusion_matrix(y_true, y_pred)
```

Construída a matriz de confusão, vamos visualizá-la. Para isso, utilizaremos o seguinte código:

```python
x_labels = ['Pred '+str(c) for c in pca_iris_df['class_codes'].unique()]
y_labels = ['True '+str(c) for c in pca_iris_df['class_codes'].unique()]
px.imshow(
        cm,
        labels=dict(x="Predictions", y="Ground Truth"),
        x=x_labels,
        y=y_labels
)
```

Para a visualização da matriz de confusão, foi necessário estabelecermos os rótulos da visualização, de predição e de valor real. Isso serve para interpretarmos a validação. Para a visualização em si, utilizamos o plotly e a função imshow, que basicamente serve para imprimir dados matriciais, desde imagens até matrizes simples.

A matriz gerada pelo scikit-learn é uma matriz quadrada i,j com tamanho igual ao número de classes que o problema tem. Sendo que as linhas (i) representam os dados reais (*grounded truth*), enquanto as colunas (j) representam as predições, como podemos ver na imagem a seguir.



Matriz de confusão. Gerado pela demonstração em Python.

Interpretando a matriz pela escala apresentada à direita da imagem, ao passar o cursor por cima dos quadrantes, podemos ver os valores exatos. Veja que a floresta aleatória funciona perfeitamente para predizer a classe 0, porém, para a classe 1 e 2, houve uma pequena confusão: o modelo errou 2 predições que foram preditas como 2, mas na realidade eram 1. Ainda assim, o resultado foi bastante satisfatório, mesmo para um exemplo didático.

Para fins de registro, extrairemos as métricas do modelo. Com o sklearn, isso pode ser feito da seguinte maneira:

```
python

f1 = f1_score(y_true, y_pred, average='weighted')
pr = precision_score(y_true, y_pred, average='weighted')
re = recall_score(y_true, y_pred, average='weighted')
accr = accuracy_score(y_true, y_pred)
metrics = {
        "f_measure": round(f1,3),
        "precision": round(pr,3),
        "recall": round(re,3),
        "accuracy": round(accr,3)
}
print(metrics)
```

Assim, teremos os resultados para apresentar tanto para pesquisa quanto para clientes.

> O dicionário de métricas deverá ter o seguinte formato: `{'f_measure': 0.935, 'precision': 0.95, 'recall': 0.933, 'accuracy': 0.933}`

Concluído o processo, basta implantar o modelo. Existem diversos modos de se fazer isso e cada um depende muito do ambiente em que será implantado. Mas, antes de qualquer coisa, precisamos salvar o modelo e, para isso, usaremos o pickle.

```
python

import pickle
# save the classifier
with open('my_dumped_classifier.pkl', 'wb') as fid:
        pickle.dump(clf, fid)
# load it again
with open('my_dumped_classifier.pkl', 'rb') as fid:
        clf_loaded = pickle.load(fid)

clf_loaded.predict(X_test[0:1])
```

O pickle persistirá o modelo no caminho declarado. Para instanciar o modelo treinado, basta fazer o pickle.load. Assim, qualquer um que tiver o arquivo pkl será capaz de rodar o modelo localmente ou na nuvem, dependendo do esquema de publicação da nuvem desejada. Existe, ainda, a alternativa com o joblib, que é praticamente igual.

Demonstração de testes e avaliação dos resultados

O especialista Fernando Cardoso Durier da Silva demonstra as atividades de testes e avaliação dos resultados.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

Questão 1

Qual estrutura auxilia os profissionais de ciência de dados a monitorarem a performance de seus modelos?

A

Árvore de decisão

B

Matriz de confusão

C

Tabela de correlação

D

Scatterplot

E

SVM

A alternativa B está correta.

A matriz de confusão é a melhor forma de avaliar os modelos de aprendizado de máquina através de métricas como a precisão, revocação (recall), medida F (F-Measure) e acurácia (accuracy).

A opção a) é um algoritmo de aprendizagem de máquina.

A opção c) não é usada em avaliação de modelos de aprendizagem de máquina.

A opção d) é um gráfico de visualização de dados.

A opção e) é um algoritmo de aprendizagem de máquina.

Questão 2

Qual teste aplica uma solução num grupo e outra ou nenhuma em outra e compara os resultados para decidir a viabilidade da solução proposta?

A

Análise de resultados

B

Matriz de confusão

C

Análise exploratória

D

Teste AB

E

Regressão linear

A alternativa D está correta.

O teste AB consiste em aplicar uma solução para um grupo/uma amostra e não aplicar nenhuma ou aplicar uma solução diferente para a outra amostra.

A opção a) não existe entre os testes disponíveis na literatura.

A opção b) é uma tabela de contingência de resultados dos modelos.

A opção c) é uma técnica do processo de mineração de dados.

A opção e) é um modelo de aprendizagem de máquina.

Considerações finais

O processo CRISP-DM é fundamental para guiar a execução e gerência de um projeto de aprendizado de máquina. Dentre as principais atividades, a coleta de dados, a limpeza e o pré-processamento dos dados são fundamentais para o sucesso do projeto de aprendizagem de máquina, visto que consomem 70% do tempo do projeto, como podemos observar pelo processo laborioso embutido nos métodos de coleta de dados.

A escolha do algoritmo de aprendizagem de máquina é orientada não só pela recomendação da literatura, mas também pela adequação do modelo aos dados e ao problema de negócio a ser resolvido. Por isso, é imprescindível que o profissional de ciência de dados converse e esteja sempre perto dos stakeholders e de especialistas do negócio. Assim, evitam-se erros de levantamento de requisitos, erros de planejamento e inconsistências dos modelos. E além da validação humana, é interessante depender, principalmente, da validação estatística garantida por meio das métricas de avaliação, bem como dos testes de hipótese.

É importante ressaltar que, por mais que o processo do CRISP-DM tenha mais afinidade com a indústria e produza artefatos como solução, é válido utilizá-lo no âmbito acadêmico, pois contempla uma fase muito importante durante a etapa de entendimento do problema, que é o levantamento de literatura para buscar trabalhos relacionados, a fim de ajudar a resolver o problema de negócio.

Podcast

Podcast

Ouça no podcast, a entrevista com o especialista Fernando Cardoso Durier da Silva sobre os assuntos abordados.

Conteúdo interativo

Acesse a versão digital para ouvir o áudio.

Explore+

Ainda que tenhamos discutido bastante sobre o projeto de aprendizado de máquina, é sempre bom se atentar ao cotidiano da área procurando por artigos recém-publicados nas bibliotecas digitais da nossa área, como IEEE, ACM, Google Scholar, Scopus, Elsevier, Springer etc., bem como fontes confiáveis, por exemplo, documentações de bibliotecas estabelecidas nas comunidades de pesquisa e desenvolvimento, como o scikit-learn, keras e tensorflow.

Nossa área é multidisciplinar, por isso vale a pena pesquisar e se aprofundar também em alguns tópicos extras. Tudo está descrito nesta lista de recomendações de pesquisa:

- Métricas de modelos
  - AUC ROC
  - Sensitivity
  - Accuracy vs Precision

- Processos de mineração de dados
    - SEMMA
    - KDD

- Estatística
    - Teste de Wilcoxon
    - Teste Chi Quadrado
    - Teste Mann Whitney

- Ferramentas de deploy de modelos
    - Data Bricks
    - Watson Machine Learning Services
    - Microsoft Azure
    - AWS

Referências

AMARAL, F. **Aprenda mineração de dados**: teoria e prática (vol. 1). Alta Books Editora, 2016.

AZEVEDO, A. I. R. L.; SANTOS, M. F. **KDD**, **SEMMA and CRISP-DM**: a parallel overview. IADS-DM, 2008.

KABIR, S. M. S. **Methods Of Data Collection**: Basic Guidelines for Research: An Introductory Approach for All Disciplines. 1 ed., p. 201-275.

SILVA, F. C. D.; GARCIA, A. C. B. **Judice Verum, a Methodology for Automatically Classify Fake, Sarcastic and True Portuguese News**. 2019.

VISA, S. *et al.* **Confusion Matrix-based Feature Selection**. MAICS, v. 710, p. 120-127, 2011.

WIRTH, R.; HIPP, J. **CRISP-DM**: Towards a standard process model for data mining. *In:* Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining. London, UK: Springer-Verlag, 2000.