Relatório 10 - Prática: Lidando com Dados do Mundo Real (II)

Yuri Vacelh Zamulhak Zdebski

Descrição da atividade

Assistir ao curso Machine Learning, Data science and generative AI with Python. As seções previstas tinham o objetivo de apresentar técnicas de data mining, data science e como lidar com dados do mundo real.

More data Mining and Machine Learning Techniques

KNN

O K-Nearest Neighbors(KNN) é um algoritmo de aprendizado supervisionado usado tanto para classificação quanto para regressão. Sua principal ideia é prever a classe ou valor de um novo ponto de dados com base nos dados mais próximos já conhecidos.

O exemplo utilizado no curso realizava classificações de filmes, a ideia era, pegar os dados de um filme e um k como entrada, em seguida calcular os K vizinhos mais próximos (comparando com todos os filmes do conjunto e guardando os k com a menor distancia).

PCA

Quando trabalhamos com dados em muitas dimensões, enfrentamos o fenômeno conhecido como maldição da dimensionalidade. À medida que o número de dimensões aumenta, o volume do espaço de dados cresce exponencialmente, fazendo com que os pontos de dados fiquem cada vez mais dispersos. Isso dificulta a identificação de padrões significativos, pois as distâncias entre os pontos se tornam menos distinguíveis.

Redução de dimensionalidade tenta destilar os dados para ordens inferiores enquanto preservam ao máximo a variância dos dados. No curso foram abordadas 2 técnicas, a K-means Clustering, que reduz os dados para K dimensões e a Principal Component Analisys (PCA), que foi a que teve maior enfase.

O PCA é uma técnica de redução de dimensionalidade que transforma dados com muitas variáveis correlacionadas em um conjunto menor de variáveis não correlacionadas, chamadas componentes principais. O objetivo é capturar a maior parte da variação presente nos dados originais, utilizando menos dimensões.

Para encontrar as componentes principais, foi utilizado o método Singular Value Decomposition, que é realmente complexo, mas a ideia principal é:

- Calcular os autovalores e autovetores dos dados
- Com base nos cálculos anteriores, definir hiperplanos
- Projetar os dados nos hiperplanos
- Escolher o numero de dimensões

O exemplo dado no curso foi utilizando o dataset Iris, um dataset que contem 150 registros de flores (no caso iris) divididos em 3 classes (Setosa, Versicolour e Virginica) e tendo 4 características a serem analisadas. A ideia foi reduzir o dataset de 4D para 2D, possibilitando assim a visualização dos dados em um scatter plot.

Data Warehousing

Os Armazéns de Dados ou Data Warehouses (DW) são grandes datacerters que tem o único proposito de armazenar grandes quantidades de dados que vem de diversas fontes. Departamentos inteiros são responsáveis em manter os DW, seja na normalização dos dados, manter a infraestrutura, backups etc.

ETL e ELT

ETL é a sigla em inglês para extração, transformação e carregamento. É a forma tradicional de se enviar dados para um DW. Os dados recém extraídos, são normalizados para os padrões dos DW, para só então serem enviados e carregados no DW, porém quando estamos lidando com um grande volume de dados (big data) outra abordagem é utilizada, a ELT.

ELT é a sigla para extração, carregamento e transformação. Ou seja, o processamento dos dados é realizado já no DW, utilizando ferramentas in-place (como hadoop) para realizar a normalização.

Reinforcement Learning

Reinforcement Learning ou aprendizado por reforço é uma técnica de machine learning que deixa o modelo explorar os dados e por meio de recompensas e punições aprende como se comportar em dado ambiente.

Q-Learning

Q-Learning é uma implementação do aprendizado por reforço, consiste em:

- conjunto de estados ambientais (s)
- conjunto de ações possíveis para cada estado (a)
- um valor para cada estado/ação (Q)

O algoritmo começa com os valores Q zerados e então começa a explorar os estados, se coisas ruins acontecem após um ação em determinado estado, o Q para essa combinação é reduzido, mas se algo bom acontece o Q aumenta.

O problema é, como explorar de forma eficiente todas essas combinações de estados e ações, algumas das opções são:

Sempre escolher os maiores valores de Q, e caso ocorra um empate escolher de forma aleatória. Essa abordagem pode pode levar a um comportamento excessivamente exploratório, negligenciando a exploração de outras ações que ainda não foram suficientemente testadas.

Introduzir um termo ϵ. Essa abordagem é um complemento da opção acima, mas antes de fazer uma decisão, o algoritmo sorteia um valor aleatório, caso ele seja menor que ϵ, a ação é escolhida de forma aleatória, caso contrario segue o raciocínio anterior. A vantagem dessa abordagem é que o agente pode continuar explorando o ambiente, evitando ficar preso em

soluções sub ótimas, ao mesmo tempo que explora as melhores opções conforme o treinamento avança.

Processos de decisão de Markov(PDS), é um framework matemático para modelagem de decisões em situações que os resultados são parcialmente aleatórios.

Programação dinâmica: Embora o Q-learning seja uma técnica de aprendizado por reforço que não exige conhecimento prévio do modelo, alguns conceitos de programação dinâmica, como a atualização das políticas de forma iterativa, podem ajudar a refinar o processo de tomada de decisão, acelerando a convergência e ajudando a explorar o ambiente de maneira mais eficiente.

O exemplo fornecido no curso foi utilizando a biblioteca Gym, utilizando o modelo do táxi. A ideia foi implementar um modelo que aprendia utilizando o aprendizado por reforço para que o táxi pegasse o passageiro e deixasse ele no destino desejado. Uma coisa que me chamou a atenção nessa implementação foi a equação para atualizar Q, que não havia sido comentada na parte teórica.

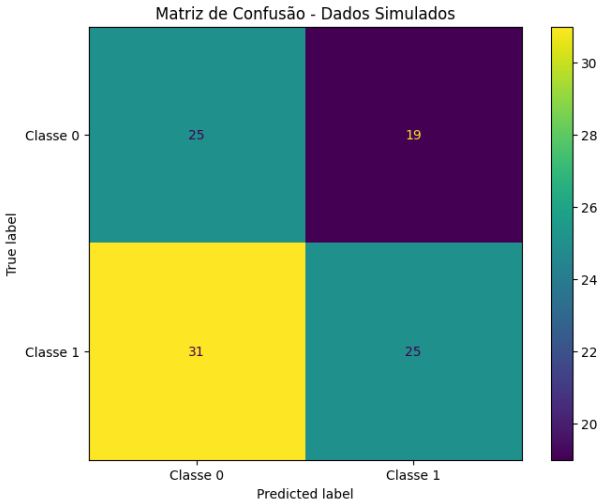
Métricas

Algumas métricas foram apresentadas, como forma de medir o desempenho de algoritmos.

Matriz de confusão

As vezes os acertos simplesmente não contam toda a historia, um teste para uma doença rara pode ter 99% de precisão apenas chutando não. Uma matriz de confusão pode ajudar, ela lista os erros e acertos das classificações de uma forma fácil e intuitiva de visualizar.

Figura 1: Matriz de Confusão



Fonte: Autoria propria

Recall

A métrica recall, também conhecida como sensitivity, true positive rate ou completeness, é uma medida de desempenho em problemas de classificação, ela avalia o percentual de positivos corretamente preditos, é utilizada quando os falsos negativos são relevantes para o problema. A fórmula para recall é dada por:

$$\text{Recall} = \frac{\text{VP}}{\text{VP} + \text{FN}}$$

Por conveniência vou utilizar as seguintes abreviações:

- VP (True Positives) são as instâncias positivas corretamente classificadas.
- VN (True Negatives) são as instâncias negativas corretamente classificadas.
- FP (False Positives) são as instâncias negativas incorretamente classificadas como positivas.
- FN (False Negatives) são as instâncias positivas incorretamente classificadas como negativas.

Precision

Também conhecida como Correct Positives é outra métrica para problemas de classificação, que avalia o percentual de resultados relevantes, é utilizada quando os falsos positivos são relevantes para o problema, a formula para precision é dada por:

$$\text{Precision} = \frac{\text{VP}}{\text{VP} + \text{FP}}$$

Specificity

Specificity é a metrica que avalia a habilidade do modelo de prever verdadeiros negativos de cada categoria disponivel, a sua formula é dada por:

$$\text{specificity} = \frac{\text{VP}}{\text{VN} + \text{FN}}$$

F1 Score

O F1-Score é a média harmônica entre a Precisão e o Recall, oferecendo um equilíbrio entre essas duas métricas. Ele é particularmente útil quando se deseja considerar tanto os falsos positivos quanto os falsos negativos, proporcionando uma medida única que reflete a performance do modelo em ambos os aspectos. Sua formula é dada por:

$$\text{F1 score} = \frac{2\text{VP}}{2\text{VP} + \text{FP} + \text{FN}}$$

RMSE

O chamado erro médio quadrático, foca na precisão do sistema, considera apenas respostas corretas e incorretas, sua formula é dada por:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

onde:

- y_i é o valor verdadeiro da i-ésima observação,
- $\hat{y}_i$ é o valor previsto para a i-ésima observação,
- n é o número total de observações.

ROC curve

É o plot do recalll pela taxa de falsos positivos, tendo varias configurações de threshold. Pontos acima da diagonal representam boas classificações.

Bias e Variância

Bias é a medida do quão distante dos valores reais a média das previsões está. A variância é uma medida do quão espalhados os valores previstos estão das respostas corretas. Esses 2 valores compõem o erro que queremos minimizar, ele tem a seguinte formula.

$$\text{Erro} = \text{Bias}^2 + \text{Variância}$$

O aumento do K no KNN aumenta a Bias e diminui a variância. Uma arvore de decisões é provável de cair no overfitting, tendo uma alta variância.

Validação cruzada

Uma forma de prevenir o overfitting é a validação cruzada. O método é dividir os dados em K partes aleatórias (os dados do conjunto, no caso), e realizar o treinamento K vezes, o treinamento ocorre com um desses K conjuntos por vez e o resto serve para teste, por fim é tirada a média das K pontuações do modelo. Esse método permite que todo o conjunto seja utilizado para o treinamento, evitando que outliers fiquem apenas nos conjuntos de teste, por exemplo.

Dealing With Real World Data

Limpeza e normalização dos dados

Como comentado na seção sobre DW, é necessário que os dados estejam em um certo padrão para as analises acontecerem de forma mais suave, essa sessão trás justamente isso, trabalhar para limpar e melhorar dados de acesso a um web site, extraindo coisas como requerimentos HTTP, mas removendo os acessos de bots, ver as paginas mais acessadas, etc.

A normalização dos dados é uma etapa importante, dependendo do algoritmo que estamos trabalhando, por exemplo outliers podem deslocar a média algumas unidades de grandeza, dependendo do range dos dados.

Feature Engineering

Features ou atributos são as características dos dados que estamos trabalhando, por exemplo no caso de identificação de escrita a mão, caixa alta e cursiva são dois exemplos de features. É importante conhecer e delimitar bem as features do problema a ser trabalhado, pois cada feature adiciona uma dimensão a mais para o problema, e consequentemente mais complexidade. Essa aula fica mais para uma reflexão, pois essas percepções vem com o tempo e com a experiencia (segundo o instrutor).

Imputing Missing Data

Dados faltantes podem influenciar em como o projeto vai se desenvolver, algumas das formas de lidar com isso são:

Substituir com a média

É um método rápido e fácil de ser implementado, porém geralmente não é o mais efetivo por perder precisão dos dados, não funcionar com dados categóricos, funciona apenas em nível de coluna, dependendo do caso é melhor utilizar a mediana (ainda tem os mesmos problemas, mas não apanha para outiliers).

Dropar a linha

Funciona para os casos de poucas linhas faltantes, porém jogar fora informação (em métodos que NECESSITAM de informação para funcionar) quase nunca é uma boa ideia.

Machine Learning

É possível utilizar métodos como KNN para encontrar os vizinhos mais próximos de uma linha e utilizar suas médias, utilizar métodos de regressão e tentar prever os valores faltantes, gerar e treinar um modelo de deep learning para gerar mais dados (dependendo do problema simplesmente inviável e como vimos em algumas LLMs pode gerar alucinações).

Coletar mais dados

Caso disponível é a melhor solução possível.

Dados desbalanceados

Quando existem discrepâncias entre os números de casos positivos e negativos, chamamos os dados de desbalanceados. Algumas das formas de lidar com isso são:

Oversampling

Simplesmente duplicar os dados da classe com menos casos, pode ser feito de forma aleatória.

Undersampling

Ao invés de duplicar, removemos os dados da maioria, porém quase nunca é prudente jogar dados fora.

SMOTE

Synthetic Minority Over-sampling TEchnique, consiste em gerar dados da classe minoria utilizando KNN (como mencionado anteriormente).

Ajustar o threshold

Ajustar a tolerância do modelo para aumentar/diminuir a sensibilidade para positivos e negativos, porém pode gerar falsos positivos/negativos.

Conclusões

O curso trouxe uma ampla visão de técnicas e conceitos utilizados no contexto de data science e machine learning, possibilitando uma maior compreensão das áreas. Por mais que fossem explicados de uma maneira mais introdutória, aumentam o leque de alternativas para quando me deparar com problemas do mundo real, direcionando para possíveis soluções.