

Introdução ao Aprendizado de Máquina

Apresentação da Disciplina



Informações iniciais

Código da disciplina

EELT7023

Nome da disciplina

Métodos Avançados em Sistemas de Energia?

Informações iniciais

Código da disciplina

EELT7023

Nome da disciplina

Métodos Avançados em Sistemas de Energia? Não será uma disciplina onde abordaremos somente exemplos da Engenharia Elétrica!

Tema abordado

Introdução ao Aprendizado de Máquina

Informações iniciais

Código da disciplina

EELT7023

Nome da disciplina

Métodos Avançados em Sistemas de Energia? Não será uma disciplina onde abordaremos somente exemplos da Engenharia Elétrica!

Tema abordado

Introdução ao Aprendizado de Máquina (faremos o possível para abordar: supervisionado, não supervisionado, redes neurais e aprendizado por reforço)

Informações iniciais

Código da disciplina

EELT7023

Nome da disciplina

Métodos Avançados em Sistemas de Energia? Não será uma disciplina onde abordaremos somente exemplos da Engenharia Elétrica!

Tema abordado

Introdução ao Aprendizado de Máquina (faremos o possível para abordar: supervisionado, não supervisionado, redes neurais e aprendizado por reforço)

OBS: Aprendizado de Máquina = Machine Learning = ML \approx IA

Informações iniciais

Código da disciplina

EELT7023

Nome da disciplina

Métodos Avançados em Sistemas de Energia? Não será uma disciplina onde abordaremos somente exemplos da Engenharia Elétrica!

Tema abordado

Introdução ao Aprendizado de Máquina (faremos o possível para abordar: supervisionado, não supervisionado, redes neurais e aprendizado por reforço)

OBS: Aprendizado de Máquina = Machine Learning = ML \approx IA

Conhecimentos prévios necessários

Por se tratar de uma disciplina introdutória, são necessários apenas conceitos básicos de matemática, estatística e programação.

Informações iniciais

Código da disciplina

EELT7023

Nome da disciplina

Métodos Avançados em Sistemas de Energia? Não será uma disciplina onde abordaremos somente exemplos da Engenharia Elétrica!

Tema abordado

Introdução ao Aprendizado de Máquina (faremos o possível para abordar: supervisionado, não supervisionado, redes neurais e aprendizado por reforço)

OBS: Aprendizado de Máquina = Machine Learning = ML \approx IA

Conhecimentos prévios necessários

Por se tratar de uma disciplina introdutória, são necessários apenas conceitos básicos de matemática, estatística e programação.

Professor

Ricardo Schumacher

Contato

Via e-mail: ricardo.schumacher@ufpr.br

- Essa disciplina cobrirá apenas **aspectos básicos** de ML, e foi concebida pensando num público que não teve contato prévio com o assunto.

- Essa disciplina cobrirá apenas **aspectos básicos** de ML, e foi concebida pensando num público que não teve contato prévio com o assunto.
- Se você já possui conhecimento nessa área (adquirido por meio de cursos online, vídeos no youtube, etc), talvez algumas explicações pareçam demasiadamente básicas

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá:

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams.

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams. Um '2', caso a envie via Teams e a mesma esteja suficientemente correta.

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams. Um '2', caso a envie via Teams e a mesma esteja suficientemente correta. Um 'P' (de Presença), caso a envie via Teams, porém, a mesma esteja demasiadamente incorreta/incompleta.

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams. Um '2', caso a envie via Teams e a mesma esteja suficientemente correta. Um 'P' (de Presença), caso a envie via Teams, porém, a mesma esteja demasiadamente incorreta/incompleta.
- Sempre que você enviar uma atividade, eu reagirei à sua mensagem.

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams. Um '2', caso a envie via Teams e a mesma esteja suficientemente correta. Um 'P' (de Presença), caso a envie via Teams, porém, a mesma esteja demasiadamente incorreta/incompleta.
- Sempre que você enviar uma atividade, eu reagirei à sua mensagem.
- A disciplina possui 30ha síncronas (15 encontros semanais de 2ha cada) + 30ha assíncronas (2ha/semana) = 60ha (4 créditos)

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams. Um '2', caso a envie via Teams e a mesma esteja suficientemente correta. Um 'P' (de Presença), caso a envie via Teams, porém, a mesma esteja demasiadamente incorreta/incompleta.
- Sempre que você enviar uma atividade, eu reagirei à sua mensagem.
- A disciplina possui 30ha síncronas (15 encontros semanais de 2ha cada) + 30ha assíncronas (2ha/semana) = 60ha (4 créditos)
- O tempo destinado à parte assíncrona da disciplina deve ser usado para assistir às aulas gravadas e realizar as atividades correspondentes

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams. Um '2', caso a envie via Teams e a mesma esteja suficientemente correta. Um 'P' (de Presença), caso a envie via Teams, porém, a mesma esteja demasiadamente incorreta/incompleta.
- Sempre que você enviar uma atividade, eu reagirei à sua mensagem.
- A disciplina possui 30ha síncronas (15 encontros semanais de 2ha cada) + 30ha assíncronas (2ha/semana) = 60ha (4 créditos)
- O tempo destinado à parte assíncrona da disciplina deve ser usado para assistir às aulas gravadas e realizar as atividades correspondentes
- Durante os encontros síncronos (sextas-feiras, das 15h30 às 17h10), tiraremos as dúvidas quanto às aulas (vídeos) e atividades da semana, via Teams

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Observações:

- Com exceção dessa aula 1, todas as demais terão “atividades de aula”. Cada aula terá uma atividade de aula correspondente
- Data limite para envio das atividades referentes a uma sexta-feira específica: 23h59 da terça-feira subsequente. Enviar via Teams. A presença em aula também será validada com base nesse envio.
- Para cada atividade, o discente receberá: Um 'F' (de Falta), caso não a envie via Teams. Um '2', caso a envie via Teams e a mesma esteja suficientemente correta. Um 'P' (de Presença), caso a envie via Teams, porém, a mesma esteja demasiadamente incorreta/incompleta.
- Sempre que você enviar uma atividade, eu reagirei à sua mensagem.
- A disciplina possui 30ha síncronas (15 encontros semanais de 2ha cada) + 30ha assíncronas (2ha/semana) = 60ha (4 créditos)
- O tempo destinado à parte assíncrona da disciplina deve ser usado para assistir às aulas gravadas e realizar as atividades correspondentes
- Durante os encontros síncronos (sextas-feiras, das 15h30 às 17h10), tiraremos as dúvidas quanto às aulas (vídeos) e atividades da semana, via Teams
- Tipicamente, teremos 2 ou 3 aulas (vídeos) por semana

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Quanto às provas:

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Quanto às provas:

- Serão realizadas de forma síncrona. Enviar o código até o final da aula, via Teams.

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Quanto às provas:

- Serão realizadas de forma síncrona. Enviar o código até o final da aula, via Teams.
- Será semelhante às atividades de aula (no sentido que, também consistirão em: completar códigos de programação, responder a perguntas presentes nos códigos, etc)

$$\text{média final} = 0.5 \times \text{atividades} + 0.2 \times \mathbf{P1} + 0.3 \times \mathbf{P2}$$

Quanto às provas:

- Serão realizadas de forma síncrona. Enviar o código até o final da aula, via Teams.
- Será semelhante às atividades de aula (no sentido que, também consistirão em: completar códigos de programação, responder a perguntas presentes nos códigos, etc)
- A disciplina não conterá qualquer tipo de "exame final"

Por meio do link abaixo:

https://www.youtube.com/playlist?list=PLaxn3fxV_3G_XVlxPa9gx_p8GDPFmjyoI

Cronograma

Início do semestre: 2/9

Cronograma previsto para os encontros:

- 6/9: Encontro 1
- 13/9: Encontro 2
- 20/9: Encontro 3
- 27/9: Encontro 4
- 4/10: Encontro 5
- 11/10: Encontro 6
- 18/10: Encontro 7 → P1
- 25/10: Encontro 8
- 1/11: Encontro 9
- 8/11: Encontro 10
- 15/11: Feriado (reposição diluída entre os dias em que a aula ultrapassa a duração prevista de 1h40)
- 22/11: Encontro 11
- 29/11: Encontro 12
- 6/12: Encontro 13
- 13/12: Encontro 14 → P2

Fim do semestre: 14/12

Bibliografia recomendada

Livros

- AURELIEN, G.. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2ed., O'Reilly Media, 2019
- RUSSEL, STUART; NORVIG, PETER. Artificial Intelligence: A Modern Approach, 3 ed., Prentice Hall, 2009.
- BISHOP, CHRISTOPHER. Pattern Recognition and Machine Learning. 2 ed., Springer, 2011.
- MOHRI, MEHYAR; ROSTAMIZADEH, AFSHIN; TALWALKAR, AMEET. Foundations of Machine Learning, 2 ed., MIT Press, 2018
- BRAGA, A.P.; LUDERMIR, T.B.; CARVALHO, A.C.P.L.F. Redes Neurais Artificiais ? Teoria e Aplicações. Rio de Janeiro: LTC, 2000
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.. Deep Learning, The MIT Press, 2016.
- THEOBALD, O.. Machine Learning For Absolute Beginners, 2 ed., Scatterplot Press, 2017.

Outros

- Vídeos e cursos online.

iniciar nova gravação?

Aprendizado de Máquina

Uma Introdução



Você sabia?

O aprendizado de máquina se faz presente quando:

- Fazemos buscas na Internet (Google, Bing, Baidu)

→ O algoritmo busca elencar quais são os resultados mais relevantes levando em conta os termos que você usou para fazer a busca.

Google search results for "Como correr uma maratona?". The search bar shows the query. Below it, there are tabs for Q, Todas, Videos (which is selected), Notícias, Imagens, Shopping, Mais, and Ferramentas. It displays approximately 106,000 results found in 0,26 seconds. The first result is a YouTube video titled "8 DICAS para CORRER MARATONA - YouTube" by "42k 8 DICAS CORRER MARATONA". The video thumbnail shows a runner and the text "42k 8 DICAS CORRER MARATONA". The description asks if the viewer is preparing for a 42k race and directs them to watch the video for 8 tips. The video was uploaded by "YouTube · Escola de Corrida de Rua · 15 de dez. de 2021". The second result is a link from "kamelturismo.com.br" to "como-treinar-para-maratona", with the text "Como treinar para uma maratona? É difícil? Veja 11 dicas". The third result is another YouTube video titled "Como treinar para uma maratona - 42km - YouTube" by "10 DICAS PARA SUA PRIMEIRA MARATONA". The thumbnail shows a man running and the text "10 DICAS PARA SUA PRIMEIRA MARATONA". The description asks if the viewer wants to know how to train for a marathon and directs them to watch the video for 10 tips. The video was uploaded by "Kamel Turismo · Programa Fôlego · 4 de jun. de 2019". The fourth result is a link from "www.youtube.com" to "watch", with the text "Importantes dicas para você saber como treinar para uma maratona com saúde, segurança e conforto. Veja essa aula...".

Você sabia?

O aprendizado de máquina se faz presente quando:

- Usamos serviços de Streaming (Netflix, HBO, YouTube, etc)
→ Levando em conta o seu perfil de consumo, qual série/filme/vídeo você provavelmente teria interesse em assistir?



Você sabia?

O aprendizado de máquina se faz presente quando:

- Solicitamos algo aos assistentes comandados por voz (Alexa, Google, Siri, etc)
→ O algoritmo é capaz de reconhecer padrões de voz etc.



Você sabia?

O aprendizado de máquina se faz presente quando:

- Quando você recebe um e-mail dizendo que ganhou R\$ 1.000.000
→ Seu gerenciador de e-mails vai provavelmente classificar esse e-mail como SPAM

✉ Parabéns!!

*  De: Anna Tamás

14 de fevereiro de 2019 17:48

Parabéns!!

O Facebook selecionou seu perfil como o vencedor do prêmio do Facebook de 2019. O prêmio vale um milhão e quinhentos mil dólares. Esta seleção foi feita pela base de dados do computador central do Facebook, onde o Facebook decidiu recompensar 10 membros do Facebook por seu uso frequente do site. Para receber seu prêmio, entre em contato com o oficial de notificação do Facebook por e-mail: facebook-postunit@outlook.com.

Saudações,

Anna Tamás
Oficial de notificação do Facebook.com

A presença do Aprendizado de Máquina em nossas vidas

Você sabia?

O aprendizado de máquina se faz presente:

- Quando você utiliza modelos de linguagem generativa (ChatGPT, por exemplo)
→ Tais modelos são treinados para entender e gerar texto de forma autônoma, com base em grandes quantidades de dados. Podem ser usados para responder perguntas, criar textos, resolver problemas, etc.

ChatGPT ▾

Qual é a temperatura máxima que um satélite orbitando a terra pode atingir quando exposto diretamente à luz solar?

A temperatura máxima que um satélite em órbita terrestre pode atingir quando exposto diretamente à luz solar depende de vários fatores, como a altitude da órbita, a superfície e o material do satélite, além de sua capacidade de dissipar calor.

No vácuo do espaço, a temperatura de um objeto exposto diretamente à luz solar pode atingir aproximadamente 120°C a 150°C (248°F a 302°F). Entretanto, a temperatura real do satélite será influenciada pelo equilíbrio térmico, que considera tanto o calor recebido do Sol quanto o calor emitido pelo satélite.

Em órbita baixa terrestre (LEO, Low Earth Orbit), por exemplo, onde muitos satélites operam, as temperaturas típicas podem variar entre aproximadamente -150°C (-238°F) na sombra da Terra e até cerca de 150°C (302°F) sob a luz solar direta.

Os satélites são projetados com sistemas de controle térmico que incluem radiadores, isolantes e, em alguns casos, circuitos ativos de resfriamento para manter os componentes em uma faixa de temperatura operacional segura, geralmente entre -40°C e 85°C.



✉ Mensagem ChatGPT



ChatGPT pode cometer erros. Considere verificar informações importantes.

Você sabia?

O Aprendizado de Máquina também encontra diversas **aplicações na indústria**:

- Simulação e modelagem matemática de sistemas
- Otimização e controle de processos industriais por meio de algoritmos inteligentes
- Monitoramento automático de sinais vitais de pacientes internados em hospitais



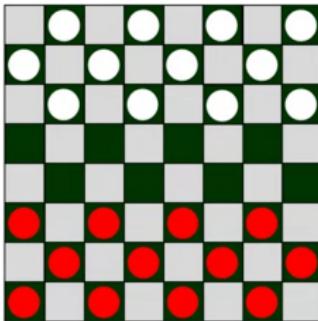
Aprendizado de Máquina = Aprendizagem de Máquina = Machine Learning = ML

Definição de Aprendizado de Máquina

Definição atribuída à Arthur Samuel (1959):

"É a área da ciência que busca fazer com que computadores sejam capazes de aprender sem terem sido explicitamente programados"

Na época, ele programou um computador para que ele aprendesse a jogar Damas de forma autônoma.



Como ele fez isso?

- Ele fez com que o computador jogasse milhares de partidas contra ele mesmo e aprendesse quais jogadas levariam à vitória (boas jogadas) e quais jogadas levariam à derrota (jogadas ruins).
- Com esse treinamento, o computador adquiriu a habilidade de jogar Damas melhor que o próprio Arthur.

Pergunta:

Se o programa de Arthur Samuel tivesse treinado com base em apenas 10 partidas de Damas contra ele mesmo, como sua performance seria afetada?

- A) Sua performance melhoraria
- B) Sua performance pioraria

Definição de Aprendizado de Máquina:

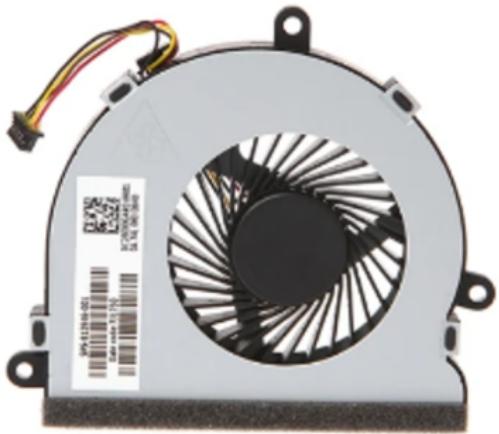
"É a área da ciência que busca fazer com que computadores (máquinas) sejam capazes de aprender sem terem sido explicitamente programados"

Definição de Inteligência Artificial (IA):

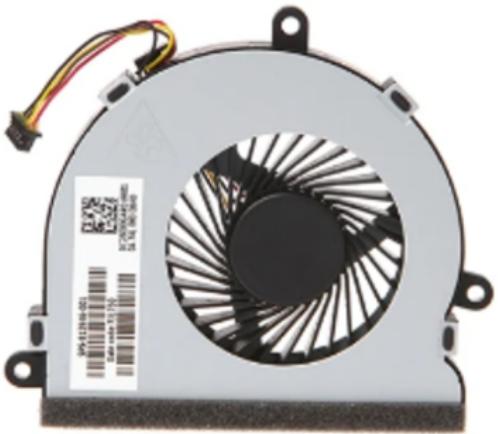
"É a área da ciência que busca fazer com que computadores pareçam ter inteligência humana"

- Em muitos casos, aprendizado de máquina é tido como sendo uma subárea da IA.
- Outras vezes, são considerados sinônimos

O Algoritmo de ML que será demonstrado agora automaticamente ajusta o valor de tensão aplicado a um Cooler para que sua velocidade de rotação [rpm] siga um determinado valor desejado.



O Algoritmo de ML que será demonstrado agora automaticamente ajusta o valor de tensão aplicado a um Cooler para que sua velocidade de rotação [rpm] siga um determinado valor desejado.



OBS: Trata-se de uma solução mais complexa em comparação à uma solução tradicional de controle usando PID, por exemplo.

A ciência de dados como um todo é uma área que apresenta hoje um nível relativamente alto de oportunidades de trabalho

Motivos:

A ciência de dados como um todo é uma área que apresenta hoje um nível relativamente alto de oportunidades de trabalho

Motivos:

- ➊ Diversas empresas conseguem alavancar fortemente seus negócios quando passam a processar dados de clientes e propor soluções inteligentes.

A ciência de dados como um todo é uma área que apresenta hoje um nível relativamente alto de oportunidades de trabalho

Motivos:

- Diversas empresas conseguem alavancar fortemente seus negócios quando passam a processar dados de clientes e propor soluções inteligentes.
- Isso é possível devido ao excesso de dados disponíveis

A ciência de dados como um todo é uma área que apresenta hoje um nível relativamente alto de oportunidades de trabalho

Motivos:

- Diversas empresas conseguem alavancar fortemente seus negócios quando passam a processar dados de clientes e propor soluções inteligentes.
- Isso é possível devido ao excesso de dados disponíveis

Observações:

- Não é simples tirar conclusões úteis a partir dos dados disponíveis
- Ensinar um computador a tirar essas conclusões de forma autônoma não é trivial.

Observação 1:

Algoritmos de Aprendizado de Máquina são como **ferramentas**. Não adianta conhecer uma ferramenta, mas não saber como usa-la apropriadamente

Observação 1:

Algoritmos de Aprendizado de Máquina são como **ferramentas**. Não adianta conhecer uma ferramenta, mas não saber como usa-la apropriadamente

Observação 2:

Por isso também falaremos bastante aqui sobre dicas de como usar de forma apropriada os algoritmos de aprendizado de máquina

Observação:

Uma vez que algoritmos de ML são ferramentas, como qualquer outra ferramenta, você pode fazer um bom uso ou um mal uso dela.

Nesse contexto, você tem dois papéis

- Papel de consumidor de algoritmos de ML → esse papel nós já exercemos faz tempo.
- Papel de desenvolvedor de algoritmos de ML

Tipos de algoritmos de Aprendizado de Máquina

Os principais tipos de algoritmos de aprendizado de máquina são:

- Algoritmos de Aprendizado Supervisionado → São o tipo de algoritmo que mais estudaremos nessa disciplina!

Os principais tipos de algoritmos de aprendizado de máquina são:

- Algoritmos de Aprendizado Supervisionado → **São o tipo de algoritmo que mais estudaremos nessa disciplina!**
- Algoritmos de Aprendizado Não Supervisionado

Os principais tipos de algoritmos de aprendizado de máquina são:

- Algoritmos de Aprendizado Supervisionado → **São o tipo de algoritmo que mais estudaremos nessa disciplina!**
- Algoritmos de Aprendizado Não Supervisionado
- Algoritmos de Aprendizado por Reforço

Os principais tipos de algoritmos de aprendizado de máquina são:

- Algoritmos de Aprendizado Supervisionado → **São o tipo de algoritmo que mais estudaremos nessa disciplina!**
- Algoritmos de Aprendizado Não Supervisionado
- Algoritmos de Aprendizado por Reforço
- Sistemas de Recomendação

Aprendizado Supervisionado

Aprendizado Supervisionado

Definição:

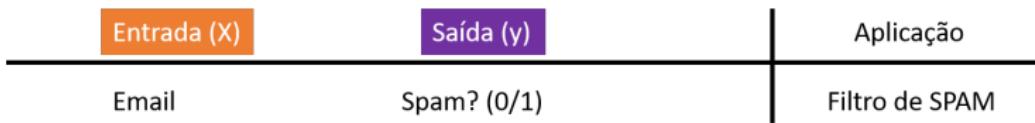
Algoritmos de Aprendizado Supervisionado são algoritmos que aprendem mapeamentos do tipo

$$X \rightarrow y$$

onde:

- X é a entrada
- y é o rótulo de saída

Exemplo:



Ideia:

Definição:

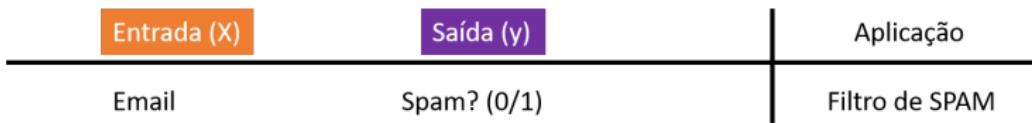
Algoritmos de Aprendizado Supervisionado são algoritmos que aprendem mapeamentos do tipo

$$X \rightarrow y$$

onde:

- X é a entrada
- y é o rótulo de saída

Exemplo:



Ideia:

- Você alimenta seu algoritmo com **diversas amostras** que contêm as "**respostas corretas**" acerca do mapeamento $X \rightarrow y$ em questão. Em outras palavras, você alimenta seu algoritmo com amostras x_i que levam aos rótulos y_i corretos.

Definição:

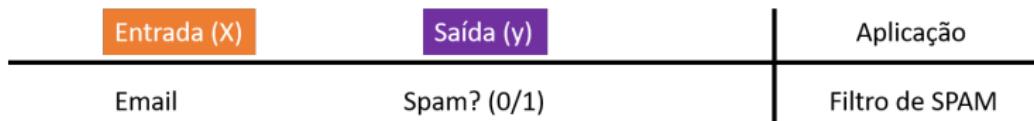
Algoritmos de Aprendizado Supervisionado são algoritmos que aprendem mapeamentos do tipo

$$X \rightarrow y$$

onde:

- X é a entrada
- y é o rótulo de saída

Exemplo:



Ideia:

- Você alimenta seu algoritmo com **diversas amostras** que contêm as "**respostas corretas**" acerca do mapeamento $X \rightarrow y$ em questão. Em outras palavras, você alimenta seu algoritmo com amostras x_i que levam aos rótulos y_i corretos.
- Sabendo que essas amostras contêm essas respostas corretas, o algoritmo buscará **aprender** qual é o padrão por trás desse mapeamento $X \rightarrow y$.

Definição:

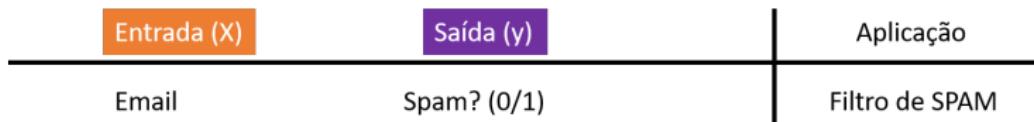
Algoritmos de Aprendizado Supervisionado são algoritmos que aprendem mapeamentos do tipo

$$X \rightarrow y$$

onde:

- X é a entrada
- y é o rótulo de saída

Exemplo:



Ideia:

- Você alimenta seu algoritmo com **diversas amostras** que contêm as "**respostas corretas**" acerca do mapeamento $X \rightarrow y$ em questão. Em outras palavras, você alimenta seu algoritmo com amostras x_i que levam aos rótulos y_i corretos.
- Sabendo que essas amostras contêm essas respostas corretas, o algoritmo buscará **aprender** qual é o padrão por trás desse mapeamento $X \rightarrow y$.
- Aprendendo esse padrão, caso ele receba uma **nova amostra específica** x_i , ele tentará **adivinhar** qual seria um rótulo apropriado para essa amostra.

Entrada (X)	Saída (y)	Aplicação
Email	Spam? (0/1)	Filtro de SPAM
Áudio	Texto transcrito	Reconhecimento de fala
Inglês	Espanhol	Algoritmo de tradução
Imagen	Posição de outros carros	Carro Autônomo
Frequências	Resposta em frequência	Eng. Elétrica
Tensão $V_{in}(t)$	Tensão $V_{out}(t)$	Eng. Elétrica
Dados de PMUs	Operação anômala? (0/1)	Eng. Elétrica

Observação:

Os algoritmos de aprendizado supervisionado podem ser de dois tipos:

- Algoritmos que realizam **Regressão**, quando y pode assumir infinitos valores distintos

Observação:

Os algoritmos de aprendizado supervisionado podem ser de dois tipos:

- Algoritmos que realizam **Regressão**, quando y pode assumir infinitos valores distintos
- Algoritmos que realizam **Classificação**, quando assume-se que y pode assumir apenas um conjunto pequeno e finito de valores

Observação:

Os algoritmos de aprendizado supervisionado podem ser de dois tipos:

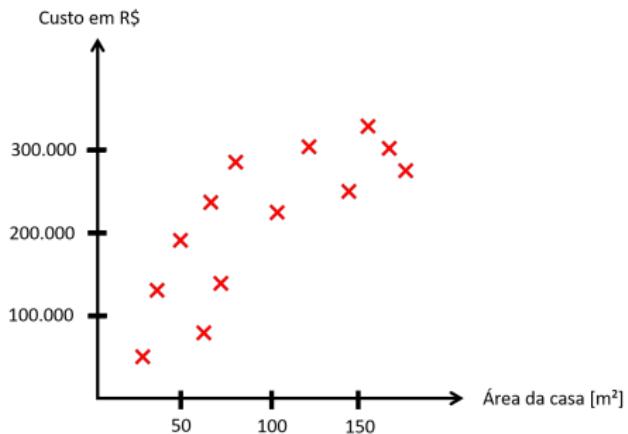
- Algoritmos que realizam **Regressão**, quando y pode assumir infinitos valores distintos
- Algoritmos que realizam **Classificação**, quando assume-se que y pode assumir apenas um conjunto pequeno e finito de valores

Pergunta: Qual tipo de algoritmo deve ser usado para o exemplo abaixo?

Entrada (X)	Saída (y)	Aplicação
Email	Spam? (0/1)	Filtro de SPAM

Regressão: Estimando o valor de casas

Você deseja estimar o valor de uma casa com base em sua área construída e, para isso, você coletou as seguintes amostras para casas à venda para o seu bairro de interesse:

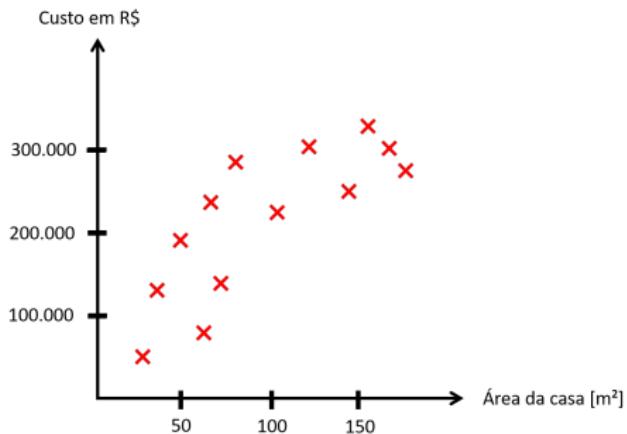


Pergunta:

Quantas amostras você coletou?

Regressão: Estimando o valor de casas

Você deseja estimar o valor de uma casa com base em sua área construída e, para isso, você coletou as seguintes amostras para casas à venda para o seu bairro de interesse:

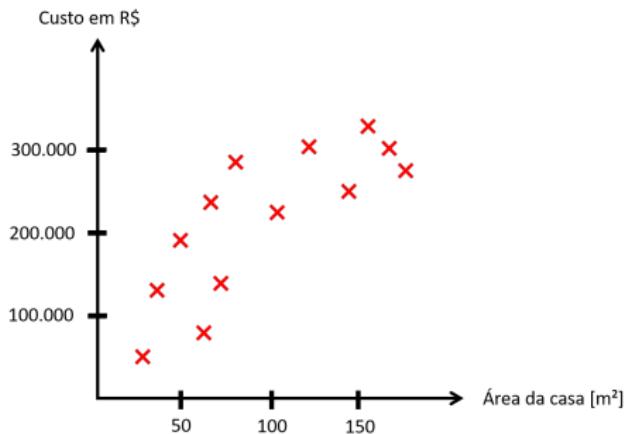


Pergunta:

O que seria um exemplo de amostra $x_i \rightarrow y_i$ para esse caso?

Regressão: Estimando o valor de casas

Você deseja estimar o valor de uma casa com base em sua área construída e, para isso, você coletou as seguintes amostras para casas à venda para o seu bairro de interesse:

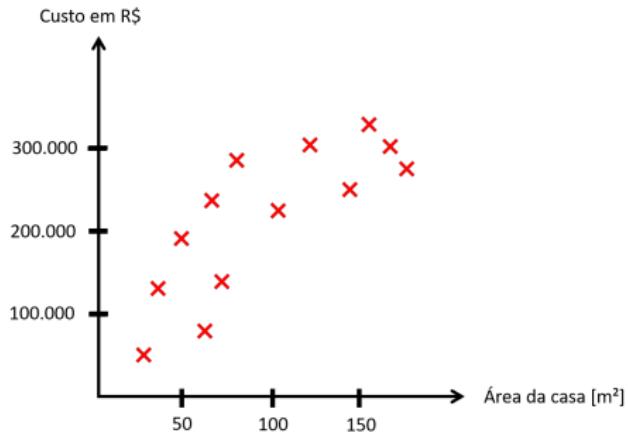


Pergunta:

Seria possível traçar uma reta que aproxima o comportamento dos dados coletados? E um modelo mais complexo?

Regressão: Estimando o valor de casas

Você deseja estimar o valor de uma casa com base em sua área construída e, para isso, você coletou as seguintes amostras para casas à venda para o seu bairro de interesse:

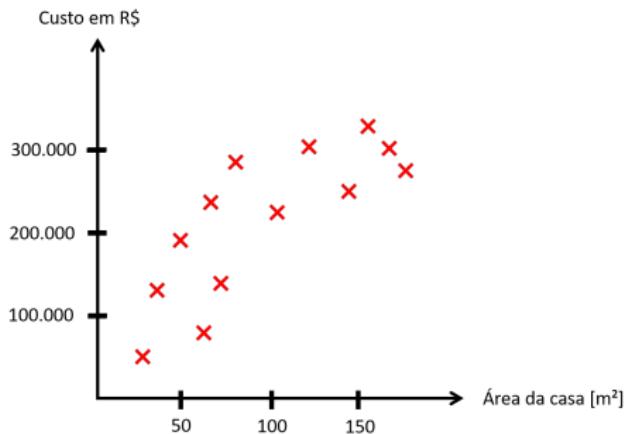


Pergunta:

Você é capaz de estimar qual seria o custo para uma casa de 50 m²? Como refinar essa estimativa?

Regressão: Estimando o valor de casas

Você deseja estimar o valor de uma casa com base em sua área construída e, para isso, você coletou as seguintes amostras para casas à venda para o seu bairro de interesse:

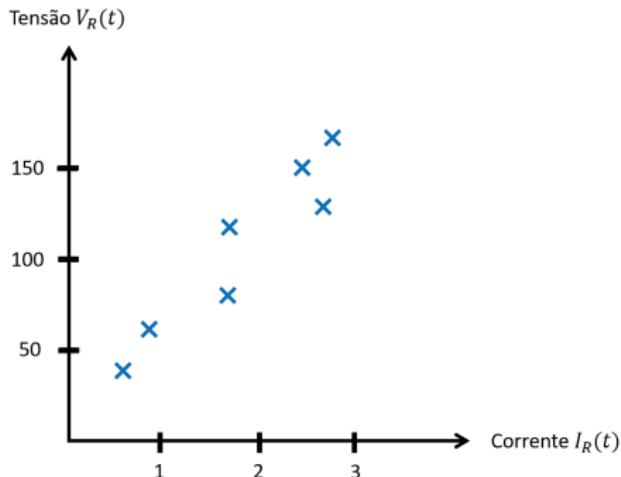


Pergunta:

Por que tal problema consiste num problema de "Regressão"?

Regressão: Estimando tensão em um resistor

Você deseja estimar o valor de tensão de um resistor com base em sua corrente e, para isso, você coletou as seguintes amostras:

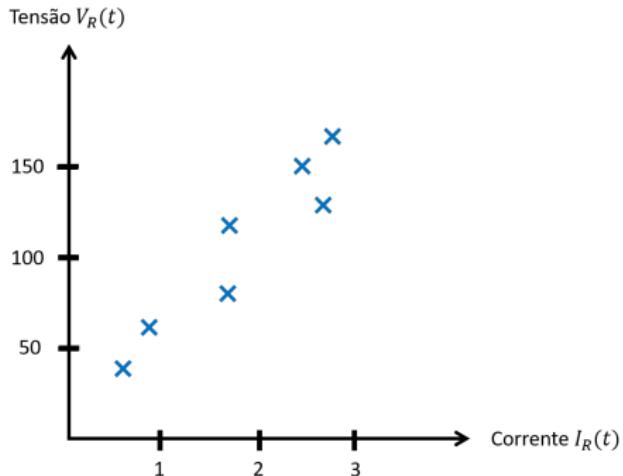


Pergunta:

Quantas amostras você coletou?

Regressão: Estimando tensão em um resistor

Você deseja estimar o valor de tensão de um resistor com base em sua corrente e, para isso, você coletou as seguintes amostras:

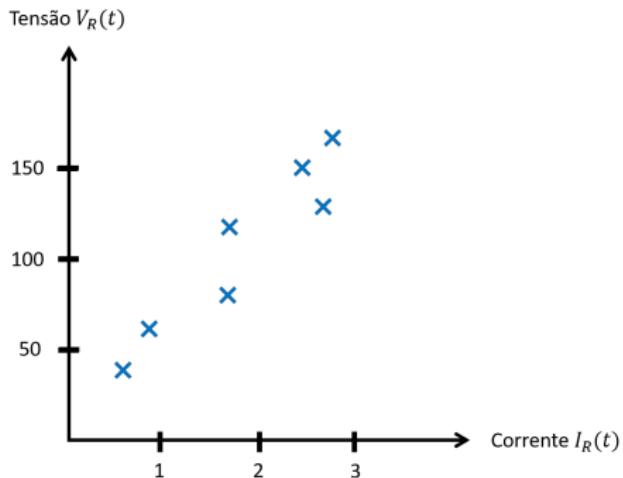


Pergunta:

O que seria um exemplo de amostra $x_i \rightarrow y_i$ para esse caso?

Regressão: Estimando tensão em um resistor

Você deseja estimar o valor de tensão de um resistor com base em sua corrente e, para isso, você coletou as seguintes amostras:



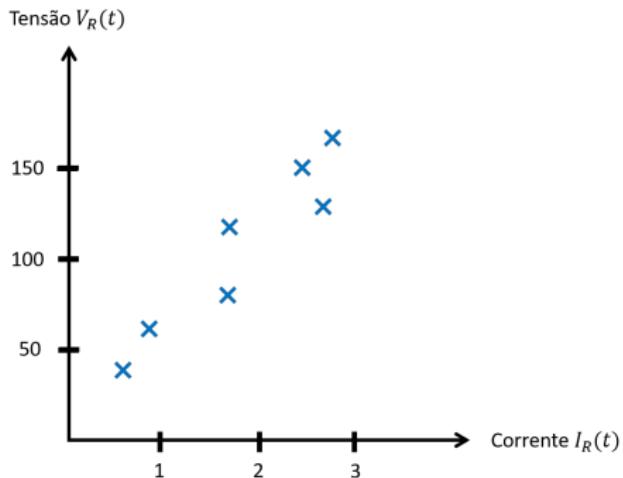
Pergunta:

Seria possível traçar uma reta que aproxima o comportamento dos dados coletados? É um modelo mais complexo?

Dica: Lembre-se da Lei de Ohm.

Regressão: Estimando tensão em um resistor

Você deseja estimar o valor de tensão de um resistor com base em sua corrente e, para isso, você coletou as seguintes amostras:

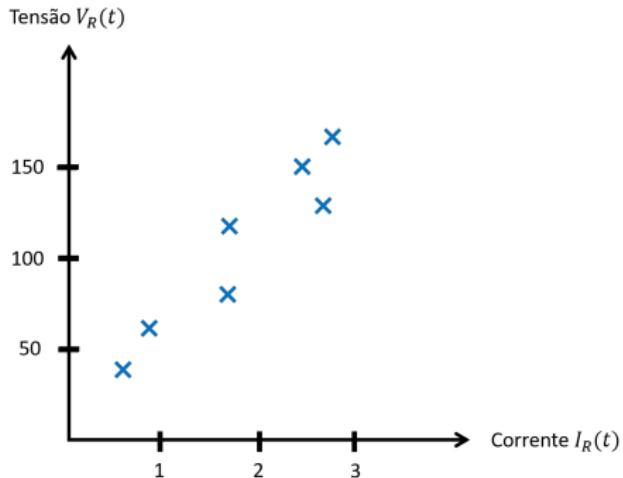


Pergunta:

Você é capaz de estimar qual seria a tensão desse resistor para uma corrente de 1.5 A? Como refinar essa estimativa?

Regressão: Estimando tensão em um resistor

Você deseja estimar o valor de tensão de um resistor com base em sua corrente e, para isso, você coletou as seguintes amostras:



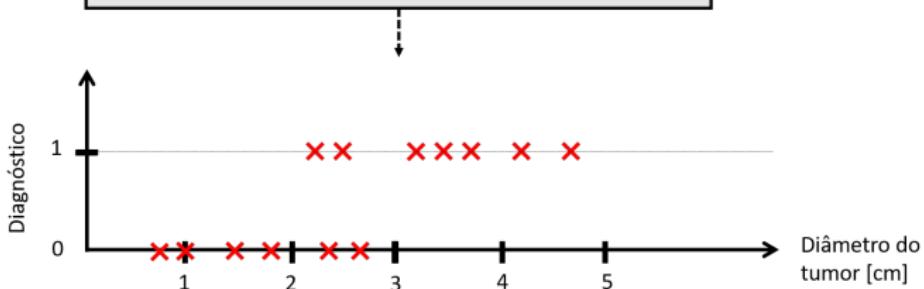
Pergunta:

Por que tal problema consiste num problema de "Regressão"?

Classificação: Diagnóstico de tumor

Você trabalha como Engenheiro de Dados para um hospital que deseja construir um algoritmo capaz de classificar se um tumor é maligno ou benigno com base em seu tamanho.

Entrada (X)	Saída (y)
0.8	0
1	0
3.2	1
4.7	1
...	



Pergunta:

Trata-se de um problema de Classificação ou Regressão?

Classificação: Diagnóstico de tumor

Você trabalha como Engenheiro/a de Dados para um hospital que deseja construir um algoritmo capaz de classificar se um tumor é maligno ou benigno com base em seu tamanho.

Entrada (X)	Saída (y)
0.8	0
1	0
3.2	1
4.7	1
...	



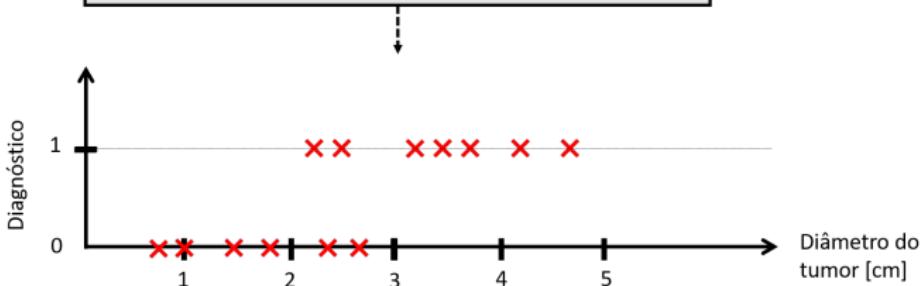
Resposta:

Na classificação, temos um valor finito de categorias possíveis para y . Nesse exemplo, quantas categorias (classes) nós temos?

Classificação: Diagnóstico de tumor

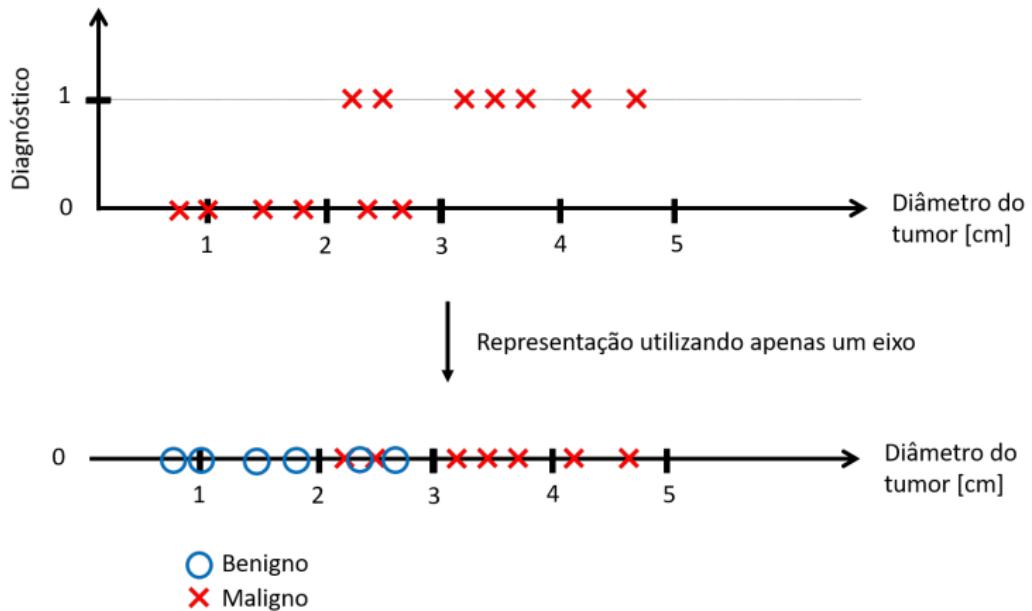
Você trabalha como Engenheiro de Dados para um hospital que deseja construir um algoritmo capaz de classificar se um tumor é maligno ou benigno com base em seu tamanho.

Entrada (X)	Saída (y)
0.8	0
1	0
3.2	1
4.7	1
...	



Pergunta:

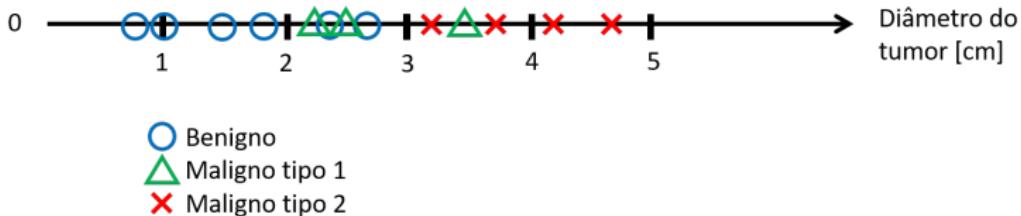
Quantas amostras temos nesse exemplo? O que seria uma amostra $x_i \rightarrow y_i$?



Observação

Para um tamanho de tumor de 2.5 cm, por exemplo, o algoritmo de ML irá retornar qual é a probabilidade desse tumor ser benigno ou maligno.

Você aprimorou o algoritmo e agora ele consegue diferenciar duas categorias diferentes de tumores malignos.

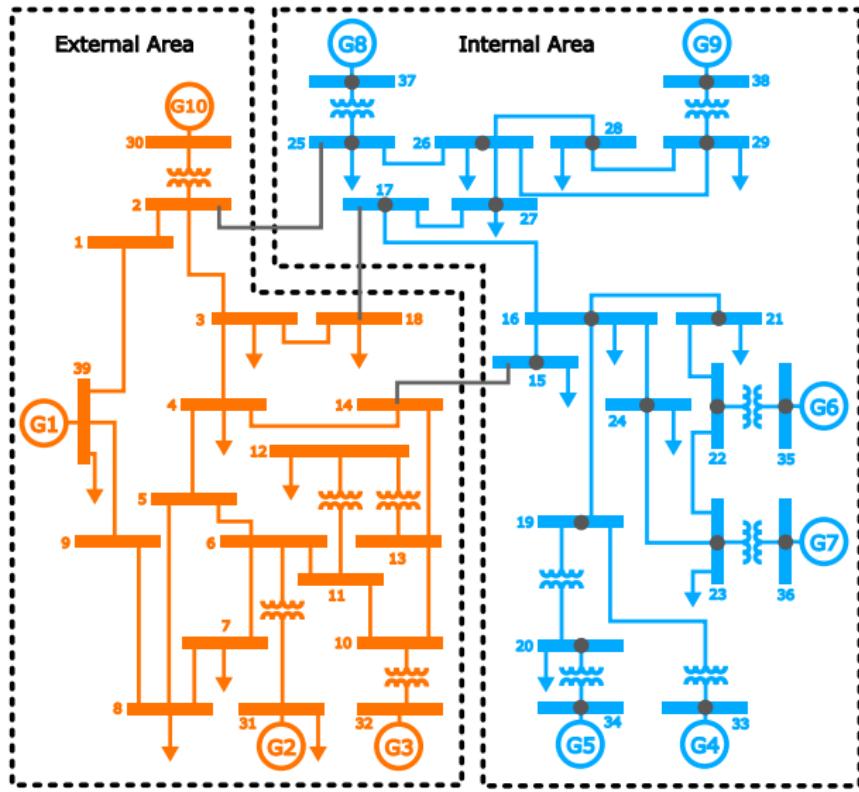


Observação

Para um tamanho de tumor de 3 cm, por exemplo, o algoritmo de ML irá estimar qual é a probabilidade desse tumor ser benigno, maligno do tipo 1, ou maligno do tipo 2.

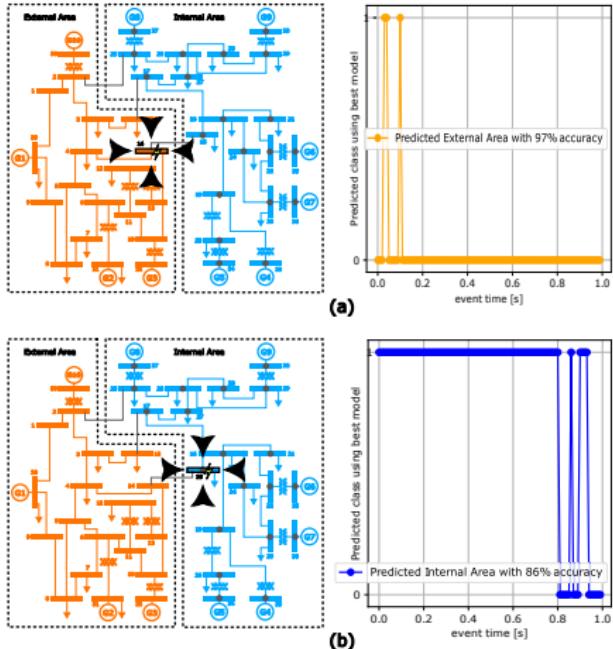
Classificação: Um exemplo na Eng. Elétrica

Localização de faltas em sistemas elétricos de geração e transmissão de energia elétrica:



Classificação: Um exemplo na Eng. Elétrica

Localização de faltas em sistemas elétricos de geração e transmissão de energia elétrica:



Algoritmos de Classificação estimam categorias (classes), ou seja, apenas um pequeno conjunto finito de valores de saída.

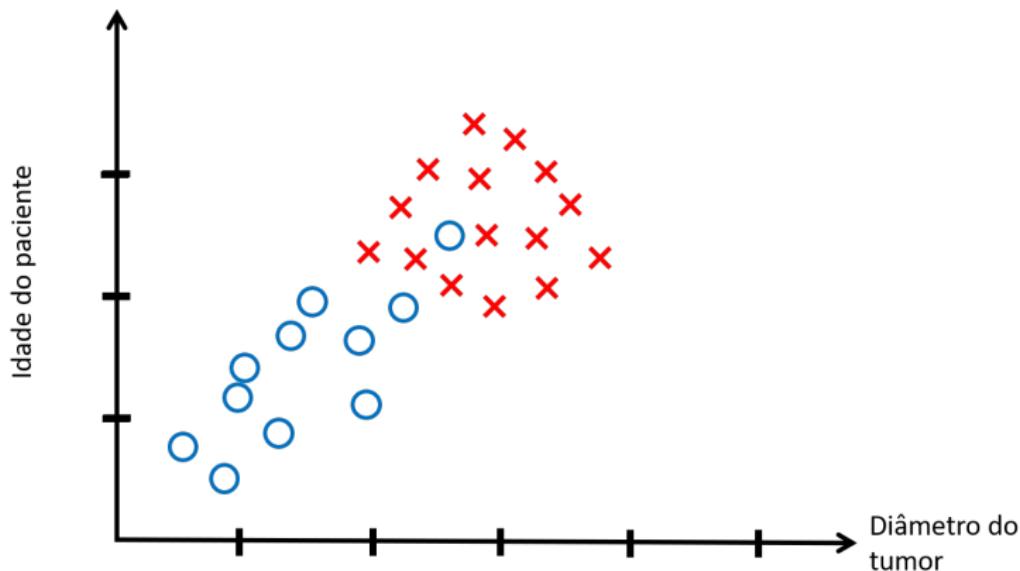
Observação

Em muitos casos, as classes não são originalmente um valor numérico:

- tumor maligno/benigno → podemos representar num formato numérico do tipo 0 ou 1.
- reconhecimento de gatos e cachorros numa imagem com os rótulos 0, 1 ou 2, considerando 0 para ausência desses animais, 1 para presença de gato e 2 para presença de cachorro.

Observações importantes

Podemos pensar num algoritmo com mais de uma entrada de dados:

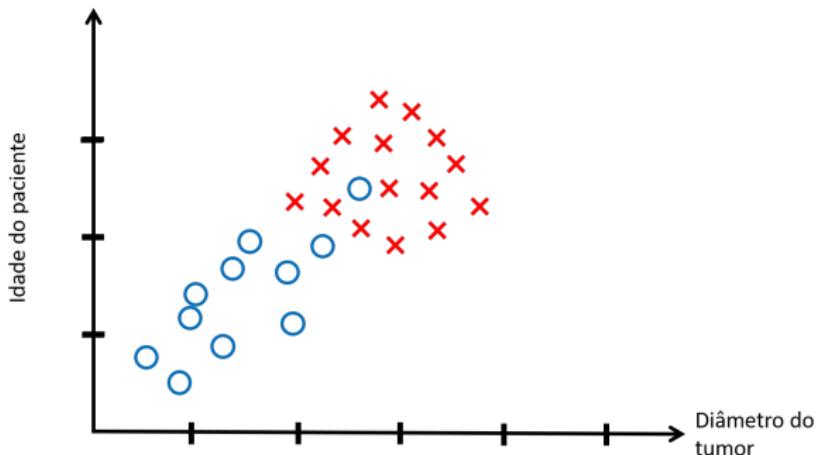


Pergunta:

Quais são as entradas desse problema? Continua sendo um problema de classificação?

Observações importantes

Podemos pensar num algoritmo com mais de uma entrada de dados:

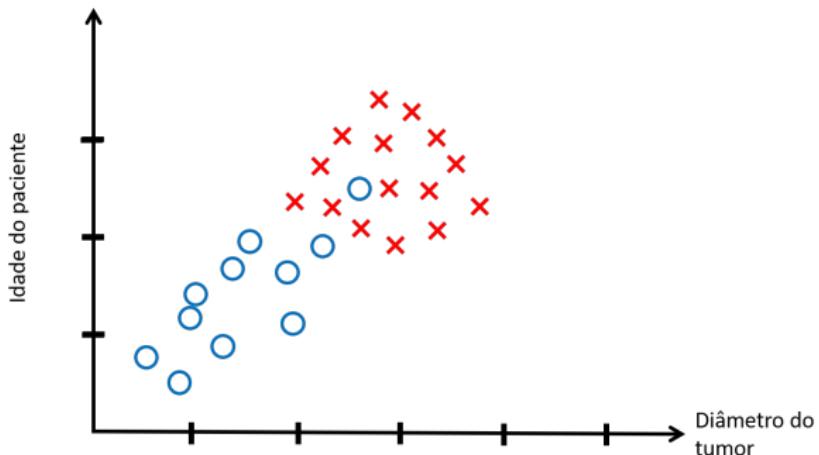


Observações:

- Nesse caso, o que um algoritmo de aprendizado pode fazer é encontrar uma curva de fronteira que separa os tumores malignos dos benignos.

Observações importantes

Podemos pensar num algoritmo com mais de uma entrada de dados:

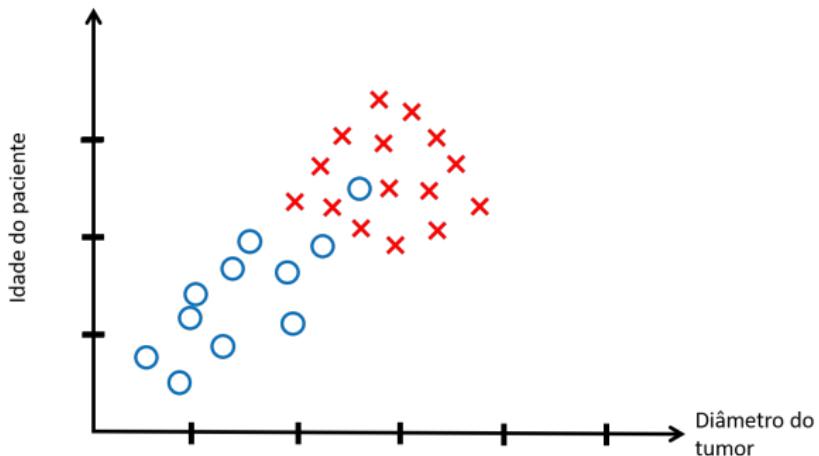


Observações:

- Nesse caso, o que um algoritmo de aprendizado pode fazer é encontrar uma curva de fronteira que separa os tumores malignos dos benignos.
- Essa curva de fronteira pode auxiliar o médico na questão de classificação do tumor para um paciente específico x_i

Observações importantes

Podemos pensar num algoritmo com mais de uma entrada de dados:



Observações:

- Nesse caso, o que um algoritmo de aprendizado pode fazer é encontrar uma curva de fronteira que separa os tumores malignos dos benignos.
- Essa curva de fronteira pode auxiliar o médico na questão de classificação do tumor para um paciente específico x_i
- Em um problema real de classificação desse tipo, geralmente um número bem maior de entradas é necessário para que o algoritmo tenha uma taxa de acerto (acurácia) significativa.

Aprendizado Supervisionado

- Considera um mapeamento do tipo entrada $X \rightarrow$ saída y

Aprendizado Supervisionado

- Considera um mapeamento do tipo entrada $X \rightarrow$ saída y
- Você alimenta seu algoritmo com **diversas amostras** que contêm as “**respostas corretas**”

Aprendizado Supervisionado

- Considera um mapeamento do tipo entrada $X \rightarrow$ saída y
- Você alimenta seu algoritmo com **diversas amostras** que contêm as "**respostas corretas**"
- O algoritmo buscará **aprender** qual é o padrão por trás do mapeamento $X \rightarrow y$ em questão.

Aprendizado Supervisionado

- Considera um mapeamento do tipo entrada $X \rightarrow$ saída y
- Você alimenta seu algoritmo com **diversas amostras** que contêm as “**respostas corretas**”
- O algoritmo buscará **aprender** qual é o padrão por trás do mapeamento $X \rightarrow y$ em questão.
- Aprendendo esse padrão, caso ele receba uma **nova amostra específica** x_i , ele poderá tentar **estimar** qual seria uma saída esperada para essa amostra.

Aprendizado Supervisionado

- Considera um mapeamento do tipo entrada $X \rightarrow$ saída y
- Você alimenta seu algoritmo com **diversas amostras** que contêm as “**respostas corretas**”
- O algoritmo buscará **aprender** qual é o padrão por trás do mapeamento $X \rightarrow y$ em questão.
- Aprendendo esse padrão, caso ele receba uma **nova amostra específica** x_i , ele poderá tentar **estimar** qual seria uma saída esperada para essa amostra.
- Algoritmos de Aprendizado Supervisionado podem ser de dois tipos principais: **Régressão** ou **Classificação**.

Régressão

Aprendizado Supervisionado

- Considera um mapeamento do tipo entrada $X \rightarrow$ saída y
- Você alimenta seu algoritmo com **diversas amostras** que contêm as “**respostas corretas**”
- O algoritmo buscará **aprender** qual é o padrão por trás do mapeamento $X \rightarrow y$ em questão.
- Aprendendo esse padrão, caso ele receba uma **nova amostra específica** x_i , ele poderá tentar **estimar** qual seria uma saída esperada para essa amostra.
- Algoritmos de Aprendizado Supervisionado podem ser de dois tipos principais: **Régressão** ou **Classificação**.

Régressão

- O algoritmo estima **um número** dentro de um conjunto com infinitos números possíveis (ex: preço de uma casa)

Aprendizado Supervisionado

- Considera um mapeamento do tipo entrada $X \rightarrow$ saída y
- Você alimenta seu algoritmo com **diversas amostras** que contêm as “**respostas corretas**”
- O algoritmo buscará **aprender** qual é o padrão por trás do mapeamento $X \rightarrow y$ em questão.
- Aprendendo esse padrão, caso ele receba uma **nova amostra específica** x_i , ele poderá tentar **estimar** qual seria uma saída esperada para essa amostra.
- Algoritmos de Aprendizado Supervisionado podem ser de dois tipos principais: **Régressão** ou **Classificação**.

Régressão

- O algoritmo estima **um número** dentro de um conjunto com infinitos números possíveis (ex: preço de uma casa)

Classificação

- O algoritmo estima **uma classe** (ex: se um tumor é benigno ou maligno)

Aprendizado Não Supervisionado

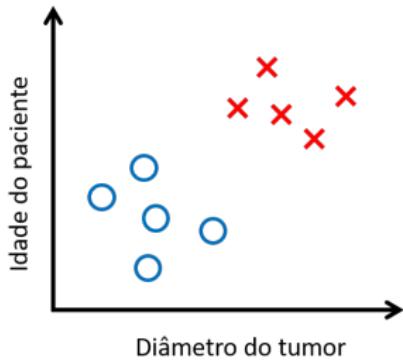
Depois do Aprendizado Supervisionado, o Aprendizado **Não** Supervisionado é o tipo de algoritmo de ML mais utilizado.

Começaremos aqui com uma comparação entre essas duas formas de Aprendizado de Máquina

Aprendizado Supervisionado *versus* Não Supervisionado

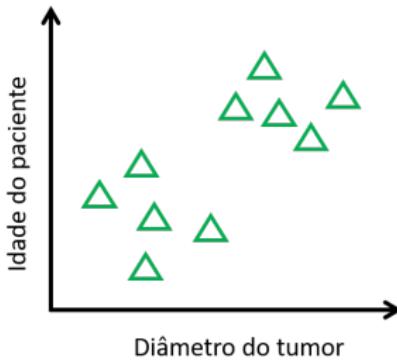
Aprendizado Supervisionado

Algoritmo aprende conhecendo
“respostas corretas”



Aprendizado Não Supervisionado

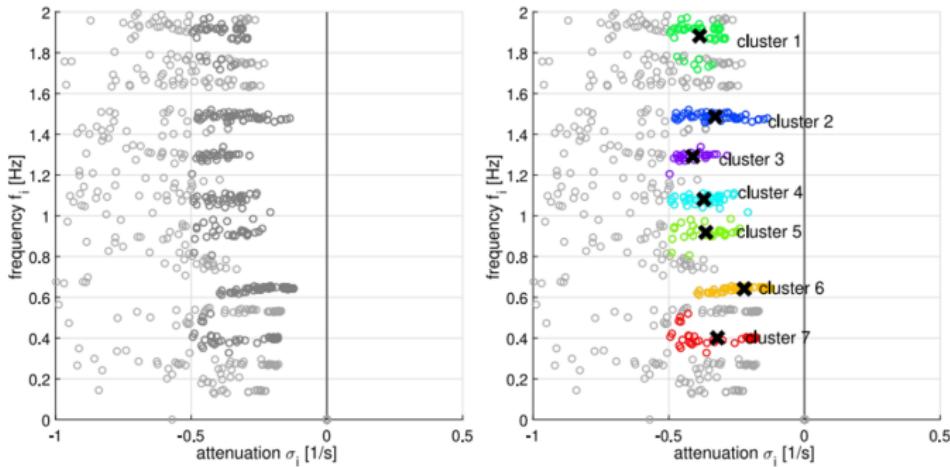
Algoritmo aprende sem conhecer as
“respostas corretas”



- No aprendizado Não Supervisionado o algoritmo busca, por ele mesmo, reconhecer padrões ou possíveis estruturas presentes nos dados
- Nesse exemplo, pode ser que o algoritmo decida que existem dois diferentes grupos de dados (*clusters*) → indicação de dois tipos diferentes de tumor
- Algoritmos de Clusterização são um tipo de Aprendizado de Aprendizado Não Supervisionado (existem diversos outros)

Um exemplo na Eng. Elétrica

Estimando oscilações presentes em sinais de Sistemas Elétricos de Potência



Pode auxiliar no processo de monitoramento em tempo real da **estabilidade** do sistema.

Link para o artigo: <https://doi.org/10.1016/j.ijepes.2021.107599>

Electrical Power and Energy Systems 135 (2022) 107599



The journal cover features the Elsevier logo (a tree and a figure) on the left, the title "International Journal of Electrical Power and Energy Systems" in the center, and a small image of power lines and a grid on the right. Below the title, it says "Contents lists available at ScienceDirect" and "journal homepage: www.elsevier.com/locate/ijepes".

Clustered R3LS: A novel approach for online estimation of power system dominant dynamics

Ricardo Schumacher*, Gustavo H.C. Oliveira, Roman Kuiava

Department of Electrical Engineering, Federal University of Paraná, 81531-980, Curitiba, Brazil



ARTICLE INFO

Keywords:

Autoregressive moving average processes
Power system monitoring
Electromechanical modes
Power system stability
Recursive estimation
K-means algorithms

ABSTRACT

Estimating electromechanical oscillations of power systems plays a crucial role to infer about their stability. In this paper it is proposed a new method in which dominant dynamics of power systems are inferred by analyzing pole estimates produced by several regularized robust recursive least squares (R3LS) implementations operating individually — each one with a different set of autoregressive moving average exogenous (ARMAX) model orders. The proposed method is named “Clustered R3LS”, since it properly adapts the so-called K-means clustering algorithm to automatically model regions of pole agglomeration produced by these individual R3LS implementations. An additional strategy for ignoring (discarding) poles identified as spurious is also part of the proposed Clustered R3LS algorithm. From a practical point of view, Clustered R3LS is here shown to more

Ideia:

- Os dados contêm apenas as entradas X , e não contêm os rótulos de saída y .
- O algoritmo busca encontrar uma **estrutura ou padrão** nos dados.

Exemplos de Algoritmos

- Clusterização (Agrupa amostras de dados com similaridade)
- Detecção de anomalias (Busca amostras de dados destoantes) → Ex: transações financeiras suspeitas
- Redução de dimensionalidade (Busca comprimir dados usando menos números)

Pergunta:

Quais dos problemas abaixo você abordaria usando um algoritmo de aprendizado não supervisionado?

- A) Dado um conjunto de artigos encontrados na Internet, agrupar eles em conjuntos de artigos com assuntos similares.

Pergunta:

Quais dos problemas abaixo você abordaria usando um algoritmo de aprendizado não supervisionado?

- A) Dado um conjunto de artigos encontrados na Internet, agrupar eles em conjuntos de artigos com assuntos similares.
- B) Dado um conjunto de e-mails rotulados como SPAM ou não SPAM, aprender um filtro de SPAM.

Pergunta:

Quais dos problemas abaixo você abordaria usando um algoritmo de aprendizado não supervisionado?

- A) Dado um conjunto de artigos encontrados na Internet, agrupar eles em conjuntos de artigos com assuntos similares.
- B) Dado um conjunto de e-mails rotulados como SPAM ou não SPAM, aprender um filtro de SPAM.
- C) Dado um conjunto de dados de consumidores, automaticamente descobrir segmentos de mercado e agrupar consumidores em diferentes segmentos de mercado.

Pergunta:

Quais dos problemas abaixo você abordaria usando um algoritmo de aprendizado não supervisionado?

- A) Dado um conjunto de artigos encontrados na Internet, agrupar eles em conjuntos de artigos com assuntos similares.
- B) Dado um conjunto de e-mails rotulados como SPAM ou não SPAM, aprender um filtro de SPAM.
- C) Dado um conjunto de dados de consumidores, automaticamente descobrir segmentos de mercado e agrupar consumidores em diferentes segmentos de mercado.
- D) Dados um conjunto de pacientes diagnosticados como tendo diabetes ou não, aprender a classificar novos pacientes como tendo diabetes ou não.

Fonte: **Machine Learning Specialization**, *deeplearning.ai*, Stanford Online, Coursera.org.

Como implementar ML na prática?

Pergunta:

Vimos que algoritmos de ML possuem diversas potencialidades. Porém, como fazemos para implementá-los na prática?

Resposta:

Fazemos isso utilizando programação de códigos.

Observação: Dentre uma quantidade bastante grande de linguagens de programação disponíveis para esse fim, nós usaremos **Python**.

Pergunta:

Como programar em Python?

Resposta:

Existem algumas possibilidades e ambientes de programação conhecidos:

- Spyder
- Jupyter Notebook
- Google Colab

Pergunta:

Como programar em Python?

Resposta:

Existem algumas possibilidades e ambientes de programação conhecidos:

- Spyder
- Jupyter Notebook
- Google Colab

Nessa disciplina, usaremos o **Google Colab** (requer conexão à Internet).

Passo-a-passo para programar usando o Google Colab

- Acesse <https://colab.research.google.com/>
- Abra um arquivo e comece a programar em Python.

Observe que os links presentes nos slides são “clicáveis”.

De olho no código!

Vamos agora ver o nosso primeiro código:

Clique no link abaixo para acessar o código:

https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula2_Introducao_Python_Notebooks.ipynb

Parte 1 (enviar código com os complementos e modificações, se necessário)

- 1) Complete o código, se necessário. → (nessa primeira atividade, note que não é necessário completar qualquer parte do código)
- 2) No código, declare novas variáveis numéricas e strings. Por meio de exemplos, também demonstre que você compreendeu a utilização de prints do tipo 'f'.

Parte 2 (enviar PDF com as respostas)

- 1) Pense num problema real onde existe a necessidade de desenvolvimento de um algoritmo de Aprendizado de Máquina.
- 2) Trata-se de um problema de aprendizado Supervisionado ou Não Supervisionado?
- 3) Caso seja um problema de aprendizado Supervisionado, trata-se de um problema de Regressão ou Classificação?
- 4) Quais serão os dados que você utilizará para alimentar o seu algoritmo?

Tópico Adicional: Instalando o Spyder e o Jupyter

Instalando o **Jupyter Notebook** ou o **Spyder** no seu PC/laptop, você poderá rodar códigos em Python localmente, sem estar conectado à Internet.

Passo-a-passo para instalação do Jupyter Notebook e do Spyder

- Instale o ambiente Anaconda no seu computador → <https://www.anaconda.com/>
- Usando o menu iniciar, procure por “Jupyter” ou “Spyder”. Abra e comece a programar em Python.

iniciar nova gravação?

Aprendizado de Máquina Supervisionado

Modelo para Regressão



Por onde começamos?

Começaremos os nossos estudos em **Modelos de Regressão** analisando o caso mais simples possível: a **regressão linear com uma variável**

Começaremos os nossos estudos em **Modelos de Regressão** analisando o caso mais simples possível: a **regressão linear com uma variável**

- A regressão linear com uma variável consiste em encontrar uma reta que modela o comportamento dos dados.

Começaremos os nossos estudos em **Modelos de Regressão** analisando o caso mais simples possível: a **regressão linear com uma variável**

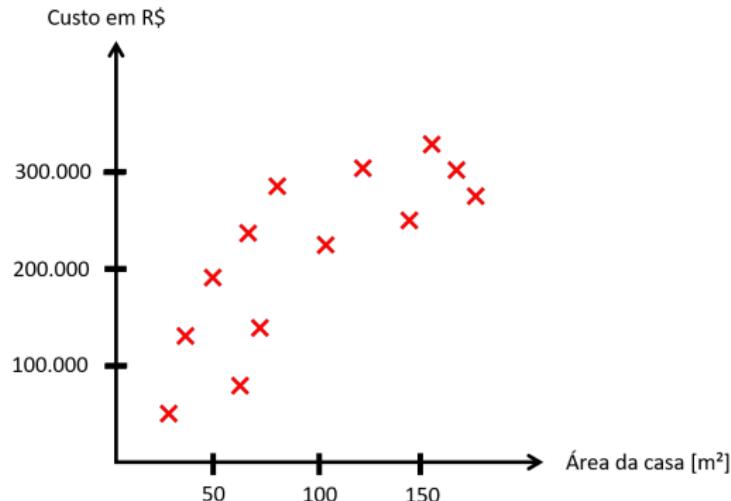
- A regressão linear com uma variável consiste em encontrar uma reta que modela o comportamento dos dados.
- Apesar de simples, tal modelagem é vastamente utilizada e pode sim ser considerar um tipo algoritmo de ML, uma vez que o modelo é diretamente treinado a partir de dados.

Por onde começamos?

Começaremos os nossos estudos em **Modelos de Regressão** analisando o caso mais simples possível: a **regressão linear com uma variável**

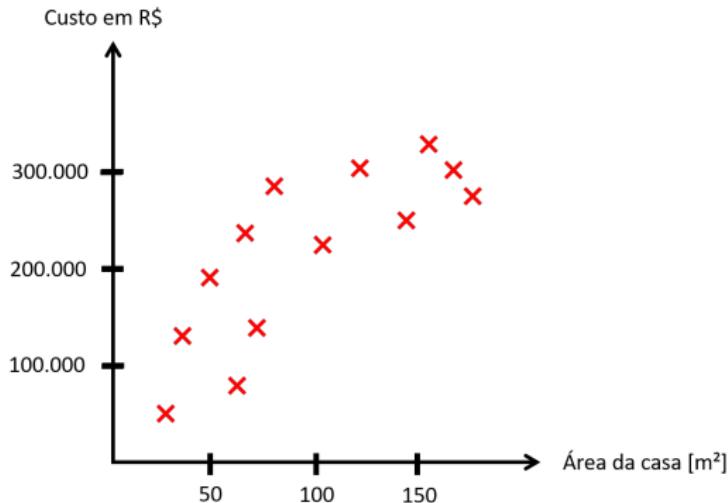
- A regressão linear com uma variável consiste em encontrar uma reta que modela o comportamento dos dados.
- Apesar de simples, tal modelagem é vastamente utilizada e pode sim ser considerar um tipo algoritmo de ML, uma vez que o modelo é diretamente treinado a partir de dados.
- Muitos conceitos que serão vistos aqui serão também aplicados nos algoritmos mais avançados vistos futuramente na disciplina

Exemplo



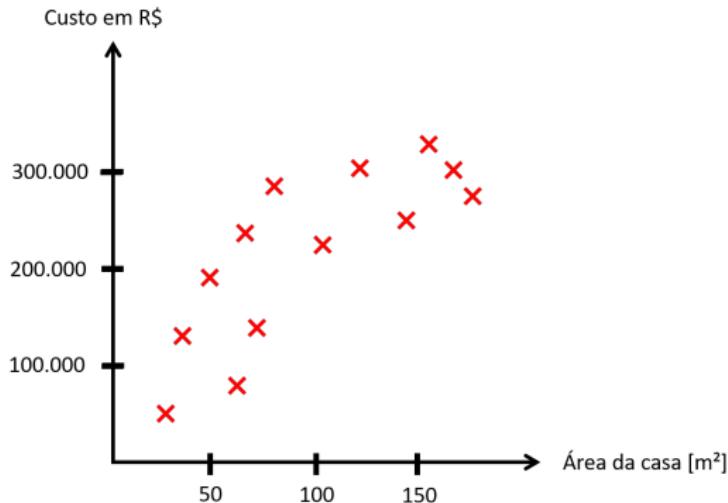
- Cada X na figura acima representa uma casa com uma certa área x_i e que foi vendida por um valor y_i

Exemplo



- Cada X na figura acima representa uma casa com uma certa área x_i e que foi vendida por um valor y_i
- Imagine agora que um amigo seu quer vender uma casa na mesma região. A casa possui 100 m². Quanto ele pode cobrar pelo imóvel?

Exemplo



- Cada X na figura acima representa uma casa com uma certa área x_i e que foi vendida por um valor y_i
- Imagine agora que um amigo seu quer vender uma casa na mesma região. A casa possui 100 m². Quanto ele pode cobrar pelo imóvel?
- Usando um algoritmo de regressão, é possível ajustar uma reta para esses dados. Essa reta poderá auxiliar o seu amigo.

Exemplo

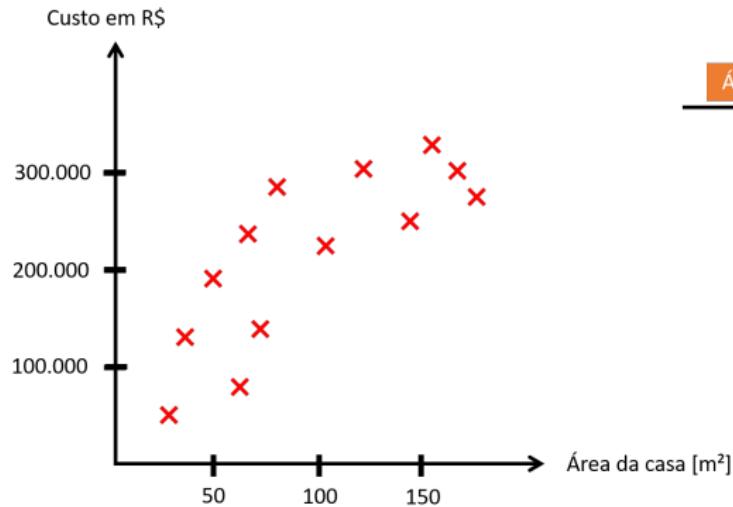


Tabela com os dados

Área da casa [m ²]	Custo em R\$
32	51.000
149	265.000
78	110.000
...	...
220	315.000

Pergunta:

Qual será o número de linhas da tabela?

Aproveitaremos esse exemplo para introduzirmos alguns termos comumente usados na área de Aprendizado de Máquina.

Conjunto de dados de treinamento:

Corresponde aos dados que serão usados para treinar o modelo. Por exemplo, a tabela toda abaixo.

Tabela com os dados

Área da casa [m ²]	Custo em R\$
32	51.000
149	265.000
78	110.000
...	...
220	315.000

Observação

Note que a casa do seu amigo não está na tabela acima. Isso porque ela ainda não foi vendida e ninguém sabe por quanto ela será de fato vendida.

Variável de entrada x

É a primeira coluna da tabela. Também chamada de **característica**, ou, **característica de entrada**. Por exemplo, para uma das amostras de treinamento da tabela, temos

$$x = 149$$

Tabela com os dados

Área da casa [m ²]	Custo em R\$
32	51.000
149	265.000
78	110.000
...	...
220	315.000

Variável de saída y

É a segunda coluna da tabela. Também chamada de **variável alvo** ou **variável de saída**. Por exemplo, para a segunda amostra de treinamento da tabela (onde $x = 149$), temos

$$y = 265.000$$

Tabela com os dados

Área da casa [m ²]	Custo em R\$
32	51.000
149	265.000
78	110.000
...	...
220	315.000

Pergunta:

O número total de amostras de treinamento “ m ” é igual ao número de linhas da tabela?

Terminologia

Amostras de treinamento

Usaremos a notação $(x^{(i)}, y^{(i)})$ para nos referirmos à i -ésima amostra de treinamento (i -ésima linha da tabela). Por exemplo, a 2^a amostra de treinamento da tabela é

$$(x^{(2)}, y^{(2)}) = (149; 265.000)$$

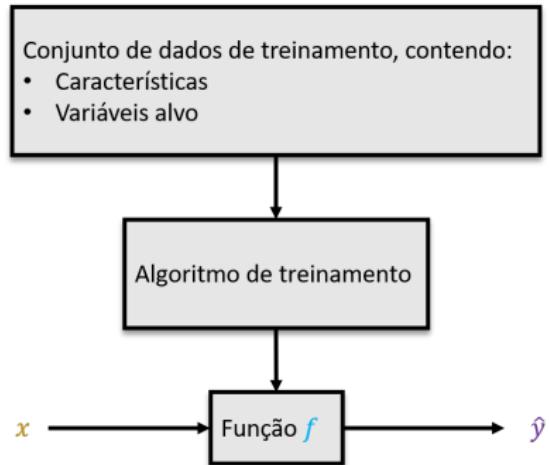
Tabela com os dados

Área da casa [m ²]	Custo em R\$
32	51.000
149	265.000
78	110.000
...	...
220	315.000

Observação

A nossa representação não significa exponenciação. Ou seja, $x^{(2)} \neq x^2$

Aprendizado Supervisionado



- A função f é também chamada de **modelo**
- Dada uma variável de entrada x , a variável \hat{y} representa uma previsão (ou, estimativa) para a variável alvo y .

Exemplo



Pergunta:

Como representar a função f ? Qual é a fórmula matemática que f irá assumir?

Resposta:

Exemplo



Pergunta:

Como representar a função f ? Qual é a fórmula matemática que f irá assumir?

Resposta:

Supondo que queremos uma reta, podemos definir

$$f_{w,b}(x) = wx + b$$

Exemplo



Pergunta:

Como representar a função f ? Qual é a fórmula matemática que f irá assumir?

Resposta:

Supondo que queremos uma reta, podemos definir

$$f_{w,b}(x) = wx + b$$

- Primeiro, precisaremos definir quanto valerá w e quanto valerá b .

Exemplo



Pergunta:

Como representar a função f ? Qual é a fórmula matemática que f irá assumir?

Resposta:

Supondo que queremos uma reta, podemos definir

$$f_{w,b}(x) = wx + b$$

- Primeiro, precisaremos definir quanto valerá w e quanto valerá b .
- Assim, para um dado valor para x , a função $f_{w,b}(x) = wx + b$ poderá ser computada.

Exemplo



Pergunta:

Como representar a função f ? Qual é a fórmula matemática que f irá assumir?

Resposta:

Supondo que queremos uma reta, podemos definir

$$f_{w,b}(x) = wx + b$$

- Primeiro, precisaremos definir quanto valerá w e quanto valerá b .
- Assim, para um dado valor para x , a função $f_{w,b}(x) = wx + b$ poderá ser computada.
- Para w e b escolhidos de forma apropriada, $f_{w,b}(x)$ nos fornecerá um valor apropriado para \hat{y}

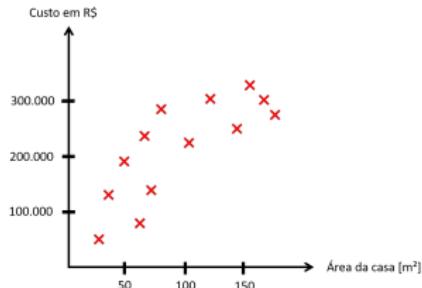
Simplificando a notação:

Em alguns casos, usaremos a notação simplificada

$$f(x) = wx + b$$

Resumo

Podemos aplicar **regressão linear com uma única variável** quando temos um conjunto de dados conforme abaixo



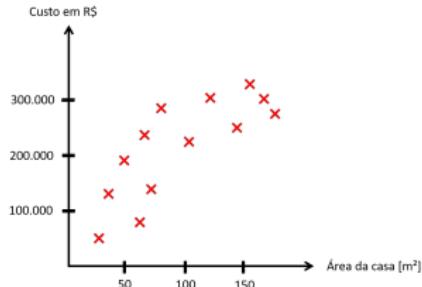
Isso significa que encontraremos um modelo do tipo

$$f(x) = wx + b$$

para esses dados.

Observações

Podemos aplicar **regressão linear com uma única variável** quando temos um conjunto de dados conforme abaixo



Isso significa que encontraremos um modelo do tipo

$$f(x) = wx + b$$

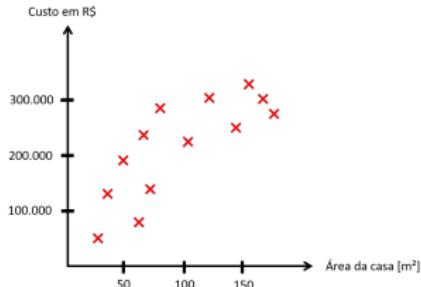
para esses dados.

Observações

- Se chama Regressão pois y pode assumir infinitos valores possíveis

Resumo

Podemos aplicar **regressão linear com uma única variável** quando temos um conjunto de dados conforme abaixo



Isso significa que encontraremos um modelo do tipo

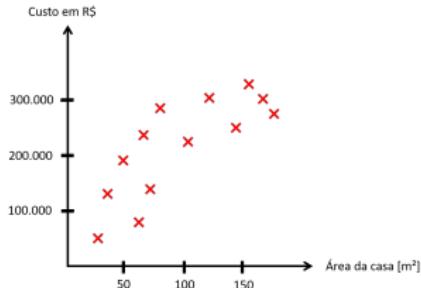
$$f(x) = wx + b$$

para esses dados.

Observações

- Se chama Regressão pois y pode assumir infinitos valores possíveis
- É linear pois não temos termos não lineares em relação aos parâmetros do modelo, ou seja, termos do tipo b^2 , w^3 , bw , \sqrt{b} , etc.

Podemos aplicar **regressão linear com uma única variável** quando temos um conjunto de dados conforme abaixo



Isso significa que encontraremos um modelo do tipo

$$f(x) = wx + b$$

para esses dados.

Observações

- Se chama Regressão pois y pode assumir infinitos valores possíveis
- É linear pois não temos termos não lineares em relação aos parâmetros do modelo, ou seja, termos do tipo b^2 , w^3 , bw , \sqrt{b} , etc.
- É uma regressão baseada em uma única variável pois temos apenas uma variável de entrada x (área da casa)

De olho no código!

Vamos agora ver no código como podemos tentar ajustar uma reta para um determinado conjunto de dados.

Clique no link abaixo para acessar o código:

https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula3_representacao_do_modelo.ipynb

Parte 1

Rode todo o código. Responda às questões nele contidas e complete-o, se necessário.

Parte 2

- 1) Pensando em como seria um conjunto de medições para um resistor de 50Ω , crie um conjunto de dados contendo 4 amostras de tensão *versus* corrente.
- 2) Insira esses dados nos códigos da Parte 1, faça as adaptações necessárias e verifique os novos resultados.

Função Custo para Regressão



Definição informal de função custo:

A função custo visa quantificar o quanto bem um modelo está se saindo ao tentar aproximar os dados.

Terminologia

função custo = função objetivo

Tabela com os dados

Área da casa [m ²]	Custo em R\$
32	51.000
149	265.000
78	110.000
...	...
220	315.000

Considere que os dados da tabela acima serão modelados por meio de uma função linear

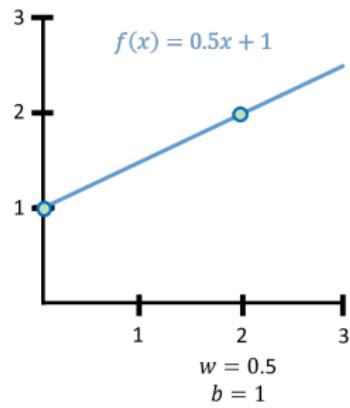
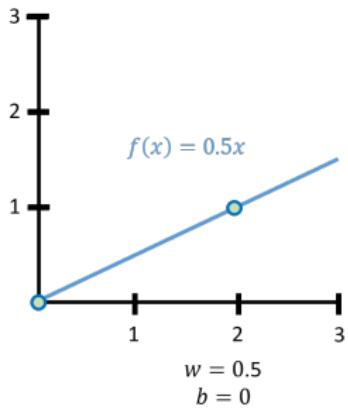
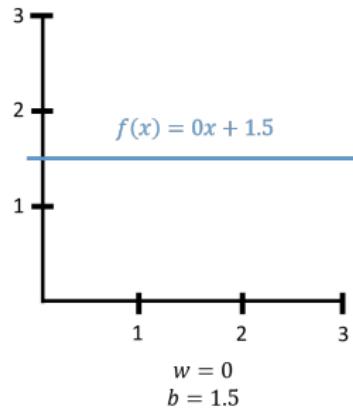
$$f(x) = wx + b$$

- w, b : parâmetros do modelo = coeficientes do modelo = pesos do modelo

"Em ML, parâmetros do modelo são as variáveis que podem ser ajustadas, durante o treinamento, com objetivo de melhorar o desempenho do modelo"

O que os parâmetros w, b definem?

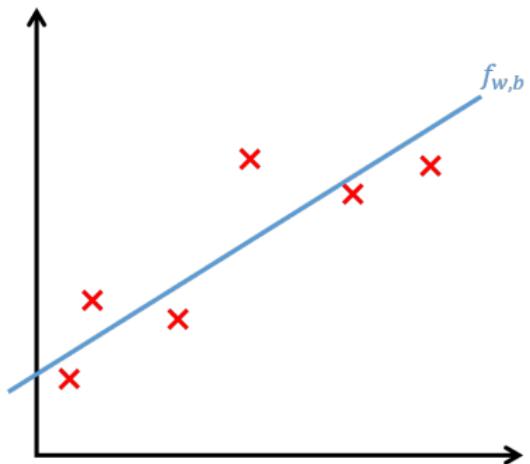
$$f(x) = wx + b$$



O que os parâmetros w, b definem?

Pergunta:

A reta abaixo aproxima bem os dados? Por que?

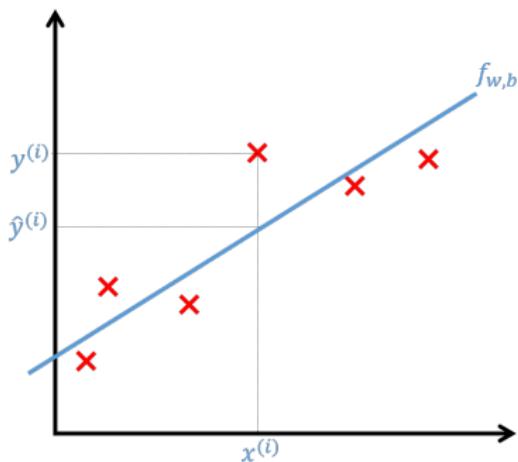


$$f_{w,b}(x) = wx + b$$

$$\hat{y} = f_{w,b}(x)$$

O que os parâmetros w, b definem?

Para uma amostra de treinamento específica, temos:



$$f_{w,b}(x^{(i)}) = wx^{(i)} + b$$

$$\hat{y}^{(i)} = f_{w,b}(x^{(i)})$$

Definindo o problema

Encontrar w, b que faz com que $\hat{y}^{(i)}$ seja próximo de $y^{(i)}$ para todas as amostras $(x^{(i)}, y^{(i)})$

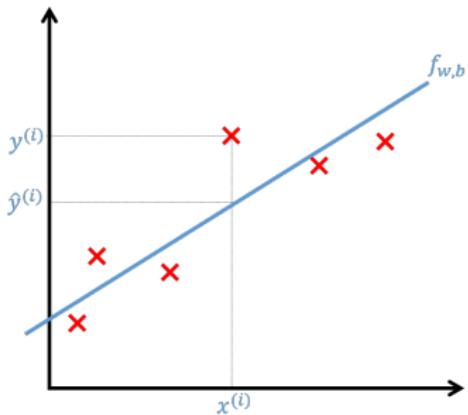
Definindo a função objetivo

Opção 1

Encontrar w, b que faz com que

$$J(w, b) = (\hat{y}^{(1)} - y^{(1)}) + (\hat{y}^{(2)} - y^{(2)}) + \cdots + (\hat{y}^{(6)} - y^{(6)})$$

assuma o menor valor possível. Lembrar que $\hat{y}^{(i)} = wx^{(i)} + b$



Pergunta:

É uma boa ideia definir J como um somatório simples de erros?

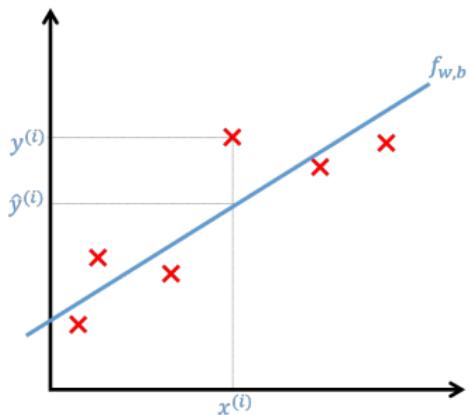
Definindo a função objetivo

Opção 2

Encontrar w, b que faz com que

$$J(w, b) = (\hat{y}^{(1)} - y^{(1)})^2 + (\hat{y}^{(2)} - y^{(2)})^2 + \dots + (\hat{y}^{(6)} - y^{(6)})^2$$

assuma o menor valor possível. Lembrar que $\hat{y}^{(i)} = wx^{(i)} + b$



Pergunta:

É uma boa ideia definir J como um somatório quadrático de erros?

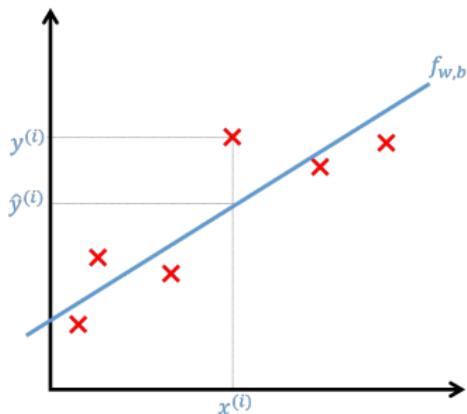
Definindo a função objetivo

Opção 3

Encontrar w, b que faz com que

$$J(w, b) = \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

assuma o menor valor possível. Lembrar que $\hat{y}^{(i)} = wx^{(i)} + b$



Pergunta:

O que foi feito aqui?

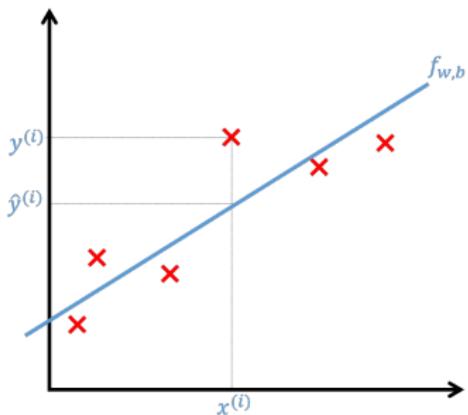
Definindo a função objetivo

Opção 4

Encontrar w, b que faz com que

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

assuma o menor valor possível. Lembrar que $\hat{y}^{(i)} = wx^{(i)} + b$



Observação

Para que J não tenda a aumentar à medida com que m aumenta, podemos fazer o chamado **Erro Quadrático Médio**.

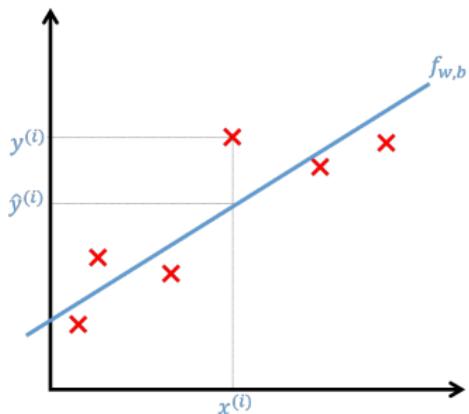
Definindo a função objetivo

Opção 5

Encontrar w, b que faz com que

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

assuma o menor valor possível. Lembrar que $\hat{y}^{(i)} = wx^{(i)} + b$



Observação

Uma divisão por "2m" ao invés de "m" também é comum em problemas de ML.

Definindo a função objetivo

Observação 1

Em problemas de regressão, pessoas irão definir diferentes tipos de função objetivo dependendo da aplicação. Entretanto, cumpre destacar que funções do tipo **mínimos quadrados** são vastamente utilizadas.

Observação 2

Nós seguiremos com a definição

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2$$

que também pode ser reescrita conforme abaixo

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m (f_{w,b}(x^{(i)}) - y^{(i)})^2$$

Buscaremos encontrar valores para w, b que fazem com que $J(w, b)$ seja o menor possível.

Pergunta:

A função custo usada na regressão linear é dada por

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Quais são os parâmetros do modelo que podem ser ajustados?

- A) w e b

Pergunta:

A função custo usada na regressão linear é dada por

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Quais são os parâmetros do modelo que podem ser ajustados?

- A) w e b
- B) $f_{w,b}(x^{(i)})$

Pergunta:

A função custo usada na regressão linear é dada por

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Quais são os parâmetros do modelo que podem ser ajustados?

- A) w e b
- B) $f_{w,b}(x^{(i)})$
- C) w , pois devemos escolher $b = 0$

Pergunta:

A função custo usada na regressão linear é dada por

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Quais são os parâmetros do modelo que podem ser ajustados?

- A) w e b
- B) $f_{w,b}(x^{(i)})$
- C) w , pois devemos escolher $b = 0$
- D) \hat{y}

Modelo:

$$f_{w,b}(x) = wx + b$$

Parâmetros:

$$w, b$$

Função custo:

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Objetivo:

$$\min_{w,b} J(w, b)$$

O que a função objetivo de fato calcula?

O que a função objetivo de fato calcula?

Para ganharmos um pouco mais de intuição acerca disso, vamos por enquanto considerar o **modelo simplificado**:

$$f_w(x) = wx \quad \rightarrow \quad \text{é como se estivéssemos considerando } b = 0$$

Ou seja, agora o problema é conforme abaixo

Modelo:

$$f_w(x) = wx$$

Parâmetro:

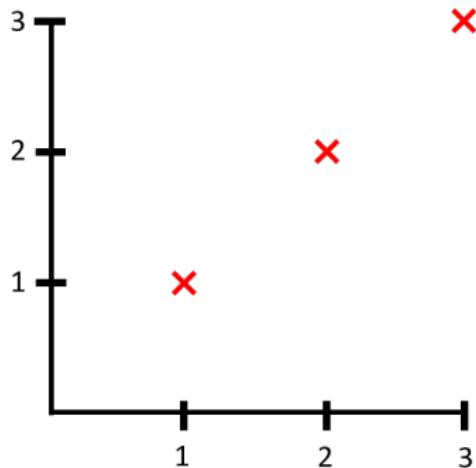
$$w$$

Função custo:

$$J(w) = \frac{1}{2m} \sum_{i=1}^m \left(f_w(x^{(i)}) - y^{(i)} \right)^2$$

Objetivo:

$$\min_w J(w)$$



Pergunta:

Para os dados acima, qual é o valor de w que resulta no melhor modelo $f_w(x) = wx$?

- A) $w = 1$
- B) $w = 0$
- C) $w = 0.5$

Exemplo

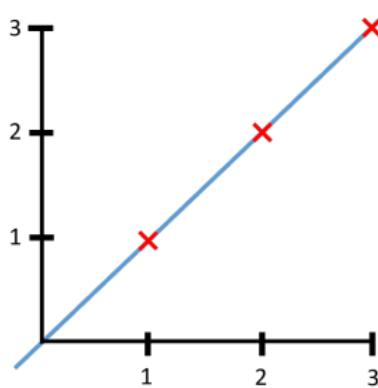
Para $w = 1$, temos

Modelo:

$$f_w(x) = wx \rightarrow f_1(x) = x \rightarrow \text{Esse modelo encontra-se representado na figura abaixo}$$

Função custo:

$$J(w) = \frac{1}{2m} \sum_{i=1}^m (f_w(x^{(i)}) - y^{(i)})^2 \rightarrow J(1) = \frac{1}{2 \cdot 3} \sum_{i=1}^3 (x^{(i)} - y^{(i)})^2 = 0$$



Pergunta: É um bom valor para w ?

Exemplo

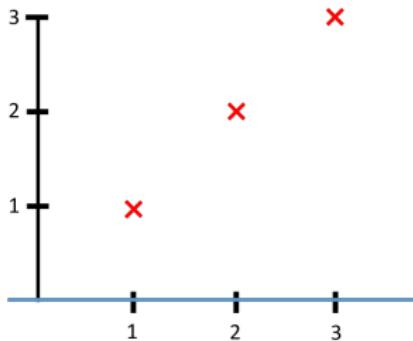
Para $w = 0$, temos

Modelo:

$$f_w(x) = wx \rightarrow f_0(x) = 0 \rightarrow \text{Esse modelo encontra-se representado na figura abaixo}$$

Função custo:

$$J(w) = \frac{1}{2m} \sum_{i=1}^m \left(f_w \left(x^{(i)} \right) - y^{(i)} \right)^2 \rightarrow J(0) = \frac{1}{2 \cdot 3} \sum_{i=1}^3 \left(0 - y^{(i)} \right)^2 \approx 2.33$$



Pergunta: É um bom valor para w ?

Exemplo

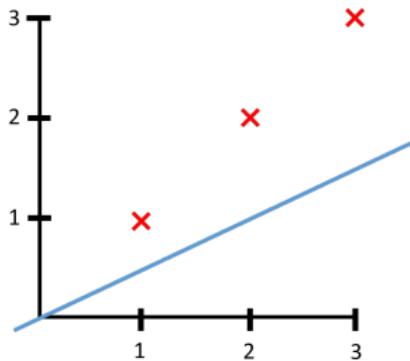
Para $w = 0.5$, temos

Modelo:

$$f_w(x) = wx \rightarrow f_{0.5}(x) = 0.5x \rightarrow \text{Esse modelo encontra-se representado na figura abaixo}$$

Função custo:

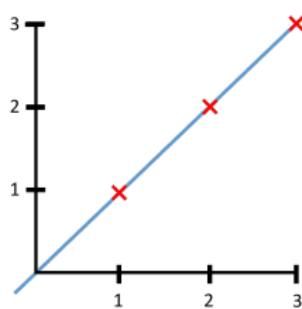
$$J(w) = \frac{1}{2m} \sum_{i=1}^m (f_w(x^{(i)}) - y^{(i)})^2 \rightarrow J(0.5) = \frac{1}{2 \cdot 3} \sum_{i=1}^3 (0.5x^{(i)} - y^{(i)})^2 \approx 0.58$$



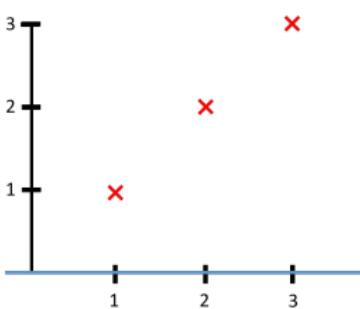
Pergunta: É um bom valor para w ?

Exemplo

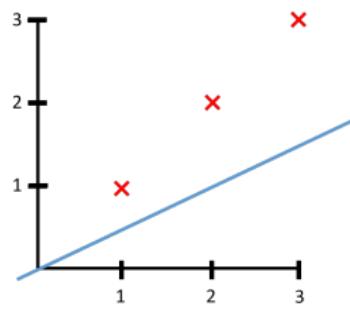
Testamos 3 valores para w , conforme resumido abaixo



$$J(1) = 0$$



$$J(0) = 2.33$$



$$J(0.5) = 0.58$$

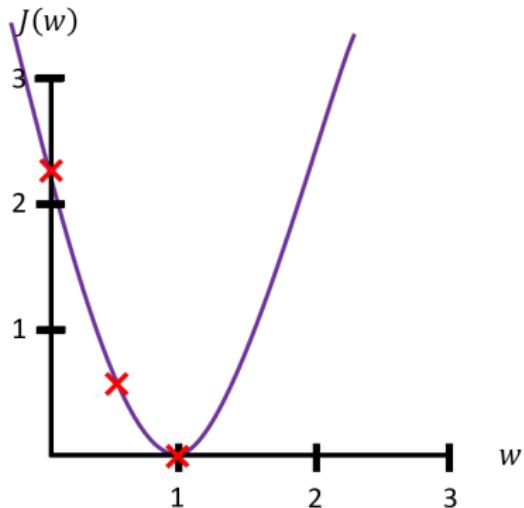
Pergunta 1:

Qual valor para w produz o menor “custo”?

Pergunta 2:

Seria possível plotar um gráfico de $J(w)$ versus w ? Qual seria sua forma?

Exemplo



Observação:

Note que agora temos um gráfico $J(w)$ versus w , e não y versus x

Pergunta:

Quando podemos considerar que o modelo se ajusta relativamente bem aos dados?

- A) Quando $f_w(x)$ está próximo de um valor mínimo para todos os valores de x presentes nos dados de treinamento.

Pergunta:

Quando podemos considerar que o modelo se ajusta relativamente bem aos dados?

- A) Quando $f_w(x)$ está próximo de um valor mínimo para todos os valores de x presentes nos dados de treinamento.
- B) Quando w é próximo de zero.

Pergunta:

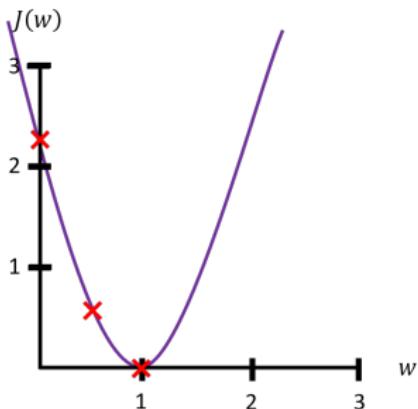
Quando podemos considerar que o modelo se ajusta relativamente bem aos dados?

- A) Quando $f_w(x)$ está próximo de um valor mínimo para todos os valores de x presentes nos dados de treinamento.
- B) Quando w é próximo de zero.
- C) Quando J está próximo de um valor mínimo ou no próprio valor mínimo.

Pergunta:

Quando podemos considerar que o modelo se ajusta relativamente bem aos dados?

- A) Quando $f_w(x)$ está próximo de um valor mínimo para todos os valores de x presentes nos dados de treinamento.
- B) Quando w é próximo de zero.
- C) Quando J está próximo de um valor mínimo ou no próprio valor mínimo.
- D) Quando x está próximo de um valor mínimo.



Escolher um valor de w que minimiza $J(w)$ parece uma boa opção! Ou seja

$$\min_w J(w)$$

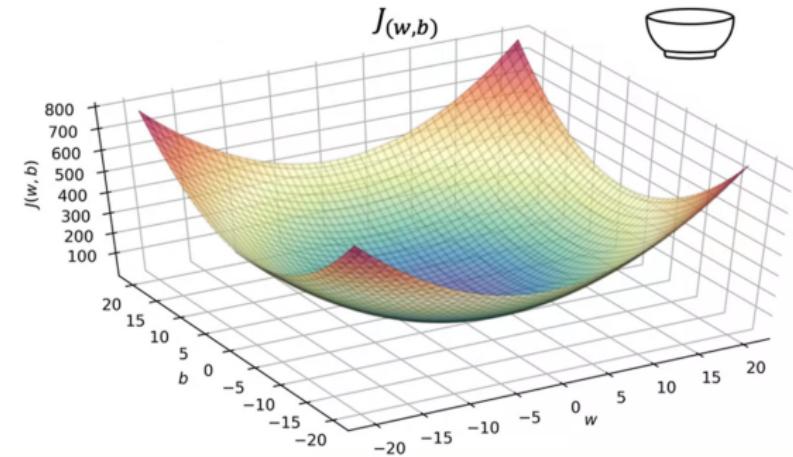
Essa ideia também é válida mesmo quando temos um modelo mais genérico:

$$\min_{w,b} J(w, b)$$

Pergunta:

Mas qual seria a forma que $J(w, b)$ tem quando temos 2 parâmetros ao invés de 1?

Como seria $J(w, b)$?



Treinando a interpretação...

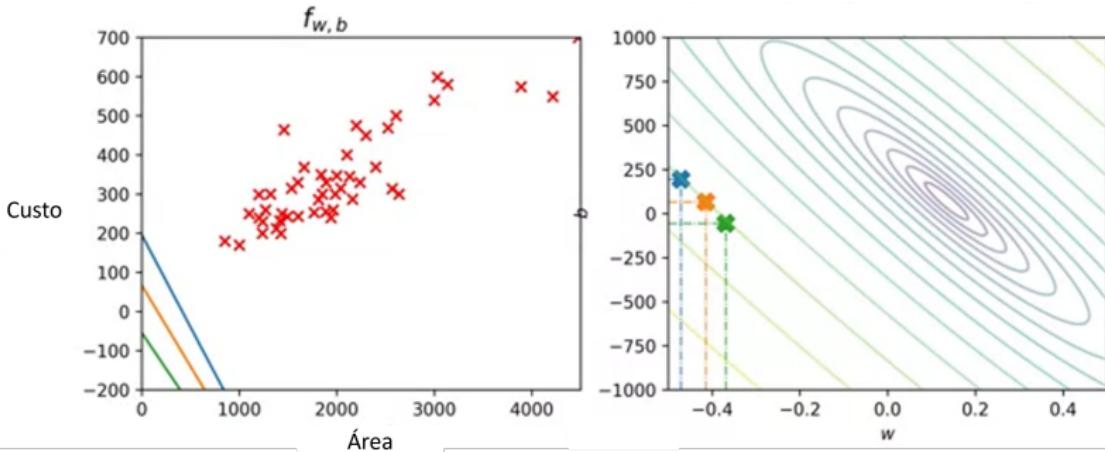
Indique no gráfico acima $J(-10, -15)$.

Pergunta:

Seria possível visualizar $J(w, b)$ como um **gráfico de contorno**?

Exemplo

Considere o exemplo abaixo onde busca-se estimar o preço de casas (preço *versus* área).

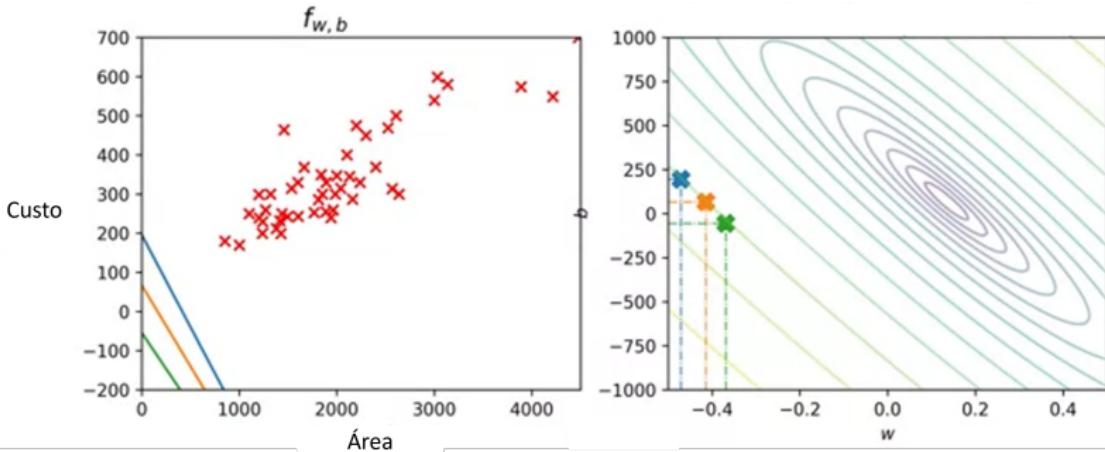


Perguntas:

- No gráfico de contorno, o que representam os 3 pontos destacados? Quais são os seus valores aproximados de b e w ? Que retas esses valores definem? Tais retas são bons modelos para os dados?

Exemplo

Considere o exemplo abaixo onde busca-se estimar o preço de casas (preço *versus* área).

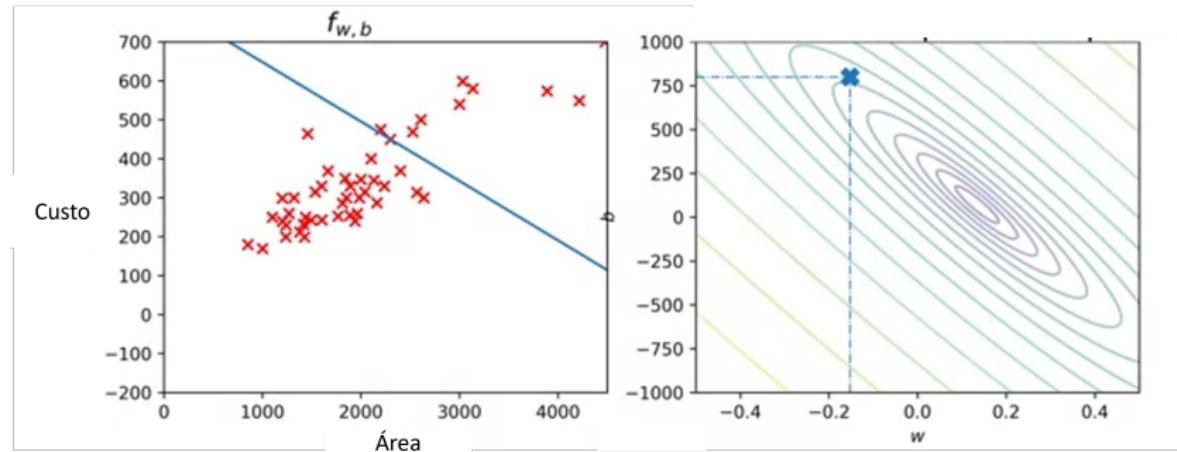


Perguntas:

- No gráfico de contorno, o que representam os 3 pontos destacados? Quais são os seus valores aproximados de b e w ? Que retas esses valores definem? Tais retas são bons modelos para os dados?
- Onde se encontra o mínimo da função $J(w, b)$? Quais são os valores aproximados para b e w nesse ponto? Que reta esses valores definem? Tal reta seria um bom modelo para os dados?

Exemplo

Visualizando o ponto $(-0.15; 800)$ no gráfico de contorno...

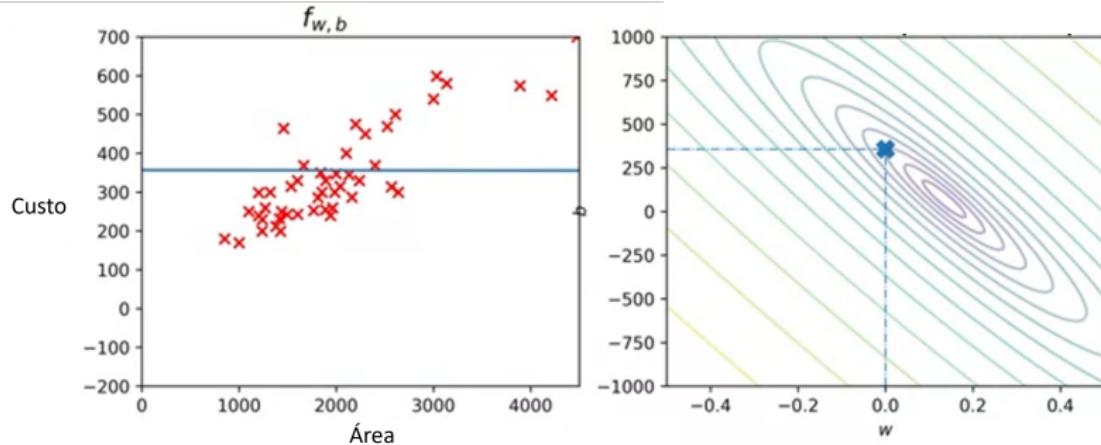


Observações

- Note que esse ponto leva a um valor de $J(w, b)$ que ainda está longe do mínimo global. Ou seja, não gera uma boa reta para representar os dados.

Exemplo

Visualizando o ponto $(0; 360)$ no gráfico de contorno...

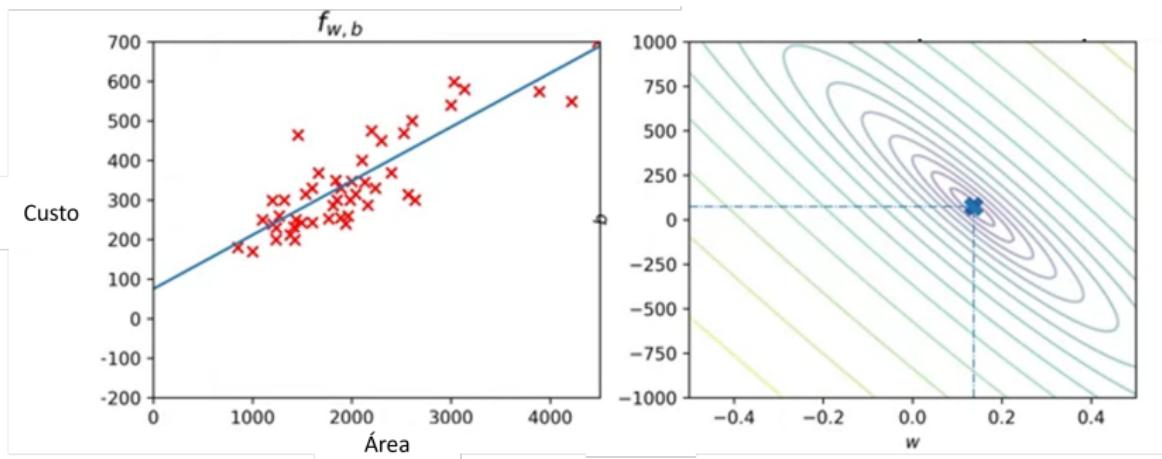


Observações

- Estamos mais próximos do mínimo de $J(w, b)$, mas ainda um pouco longe do mínimo global.

Exemplo

Visualizando o ponto $(0.13; 71)$ no gráfico de contorno...



Observações

- Note que agora estamos praticamente no mínimo de $J(w, b)$
- Isso significa que as distâncias verticais existentes entre $f(x)$ e y foram minimizadas

De olho no código!

De olho no código!

Veremos agora como representar em código a nossa função custo.

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximpvp2/master/blob/main/codigo_aula4_funcao_custo.ipynb

Parte 1

Rode todo o código. Responda às questões nele contidas e complete-o, se necessário.

Parte 2

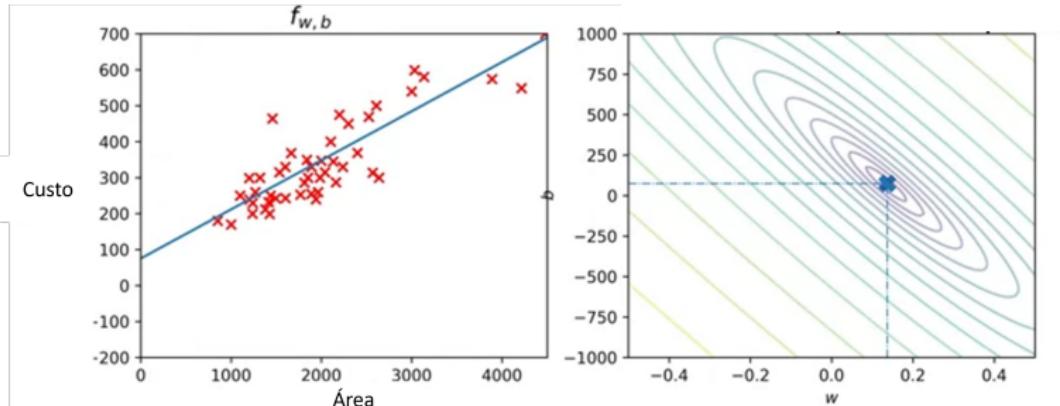
Insira no código da Parte 1 o conjunto de medições que você já criou anteriormente para um resistor de 50Ω , faça as adaptações necessárias e verifique os resultados.

Minimizando a função custo pelo gradiente



Onde estamos e para onde vamos?

Na aula anterior, vimos que minimizando a função custo $J(w, b)$ nós conseguimos obter um modelo que minimiza a soma dos erros quadráticos $\left(f_{w,b}(x^{(i)}) - y^{(i)}\right)^2$, onde $f_{w,b}(x^{(i)})$ denota a previsão feita pelo modelo.



Pergunta:

Seria possível programar um algoritmo que busca automaticamente os parâmetros do modelo w, b que minimizam a função custo?

Resposta:

Sim. Nós faremos isso agora usando o **Método do Gradiente**.

Definição informal:

O **Método do Gradiente** consiste numa forma sistemática de busca por parâmetros (valores numéricos) que minimizam uma dada função.

Definição informal:

O **Método do Gradiente** consiste numa forma sistemática de busca por parâmetros (valores numéricos) que minimizam uma dada função.

No nosso caso...

No nosso caso, aplicaremos o **Método do Gradiente** para buscar sistematicamente os valores de w e b que minimizam a função custo

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Observações importantes:

O Método do Gradiente:

- pode ser usado para minimizar diversos tipos de funções (não precisa ser necessariamente uma função custo), e não se limita apenas a dois parâmetros.
- é vastamente utilizado na área de Aprendizado de Máquina, desde algoritmos mais simples (como o nosso, por enquanto), até algoritmos altamente avançados e sofisticados, como redes neurais profundas, etc.

No nosso caso...

Temos a função custo

$$J(w, b)$$

e queremos

$$\min_{w,b} J(w, b)$$

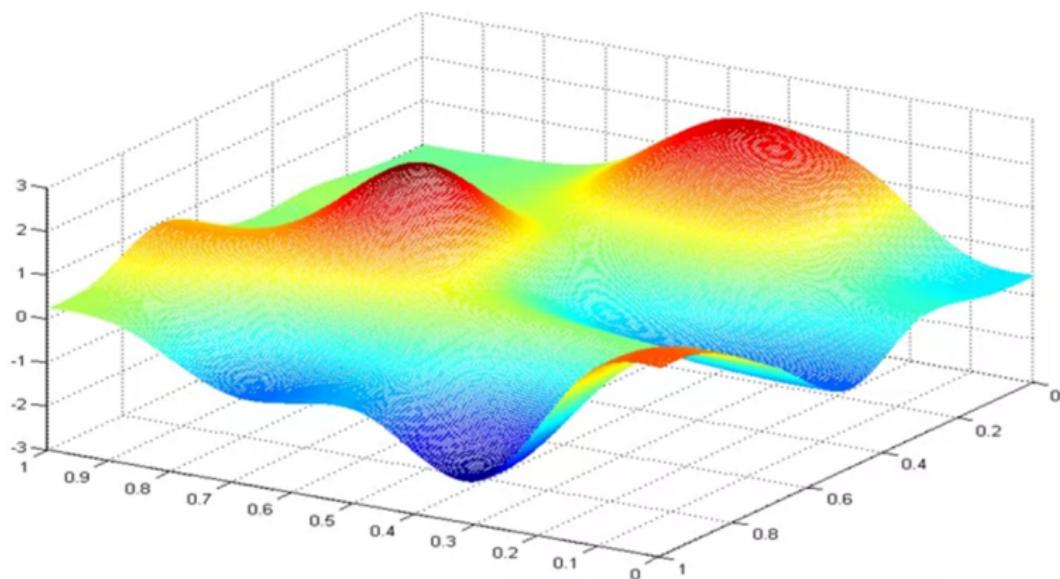
Observação

O Método do Gradiente:

- pressupõe um palpite inicial para os parâmetros. Ou seja, teremos que iniciar o algoritmo com valores para w , b . Por exemplo, podemos começar com $w = b = 0$.
- altera os valores de w , b com o objetivo de reduzir $J(w, b)$ até que estejamos próximos de um valor mínimo

Observação importante

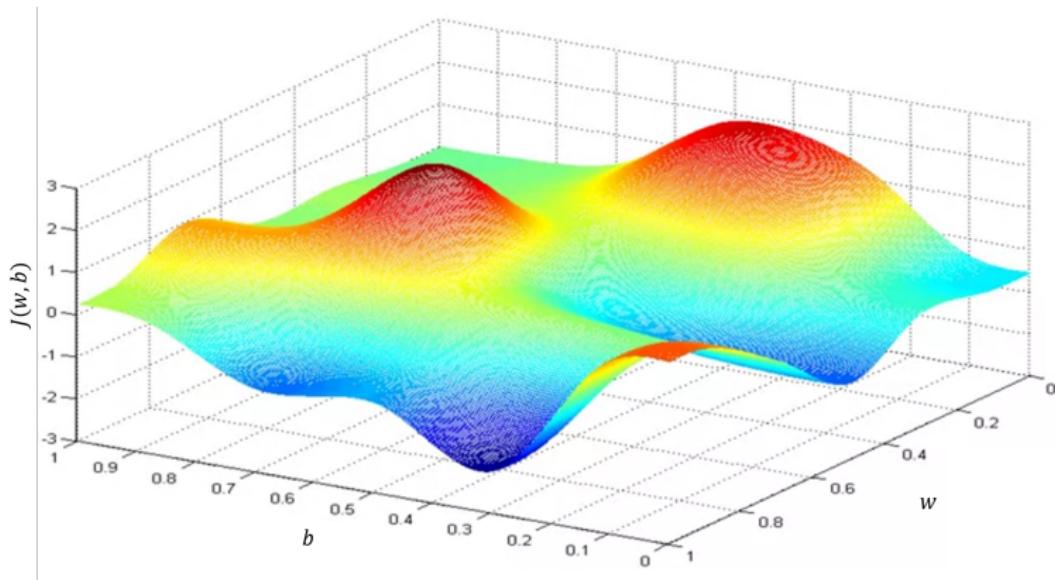
Funções mais complexas podem ter mais do que 1 mínimo.



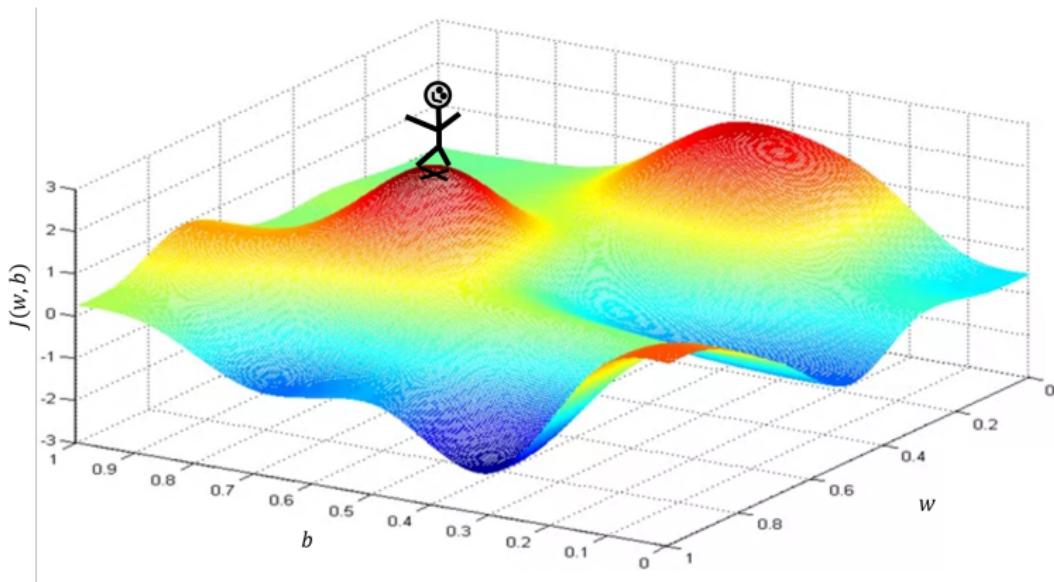
Observação

Esse tipo de função mais complexa é obtida quando treinamos modelos mais complexos, como redes neurais, por exemplo.

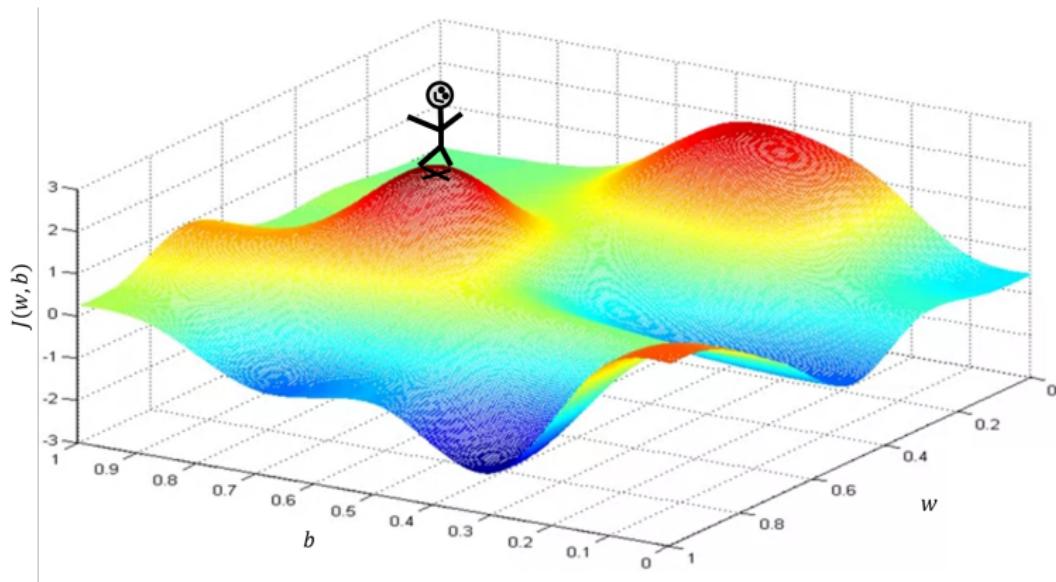
Suponha agora que os eixos são w , b e $J(w, b)$.



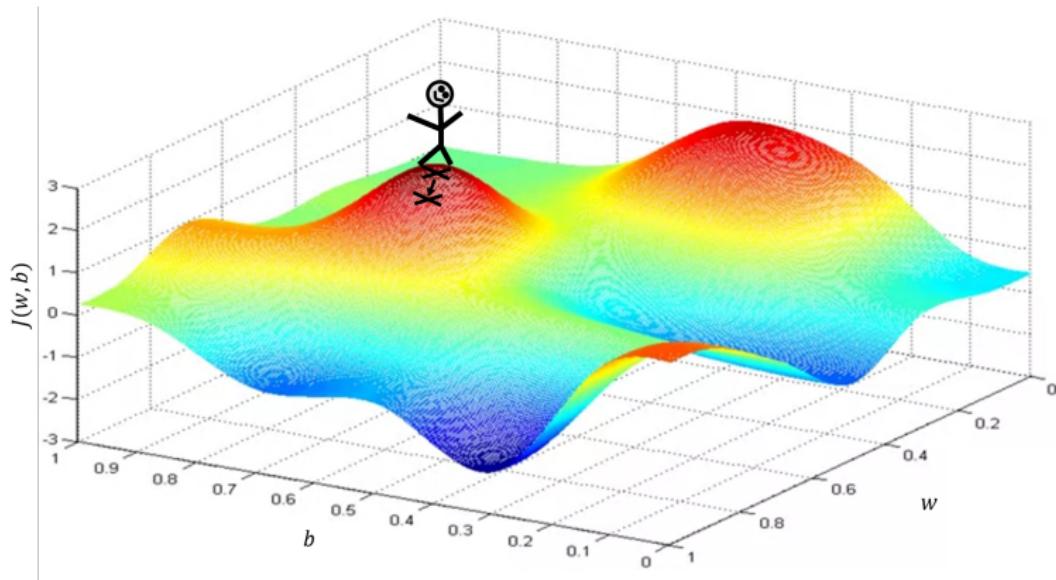
Você está numa posição específica dessa montanha e deseja chegar no seu ponto mais baixo.



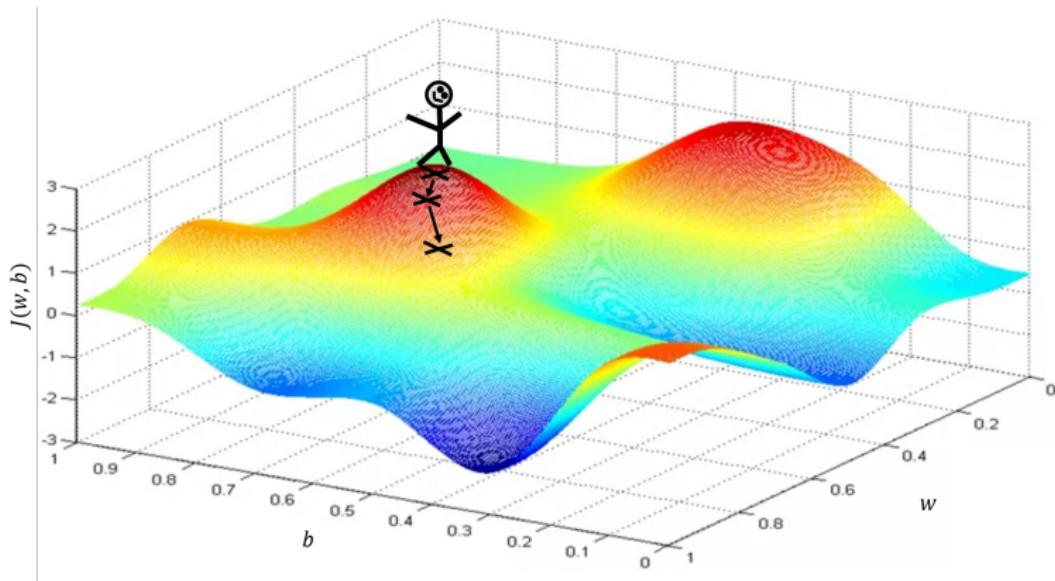
Você olha ao redor e tem que decidir para qual lado você dará o seu próximo passo. Para qual direção você iria?



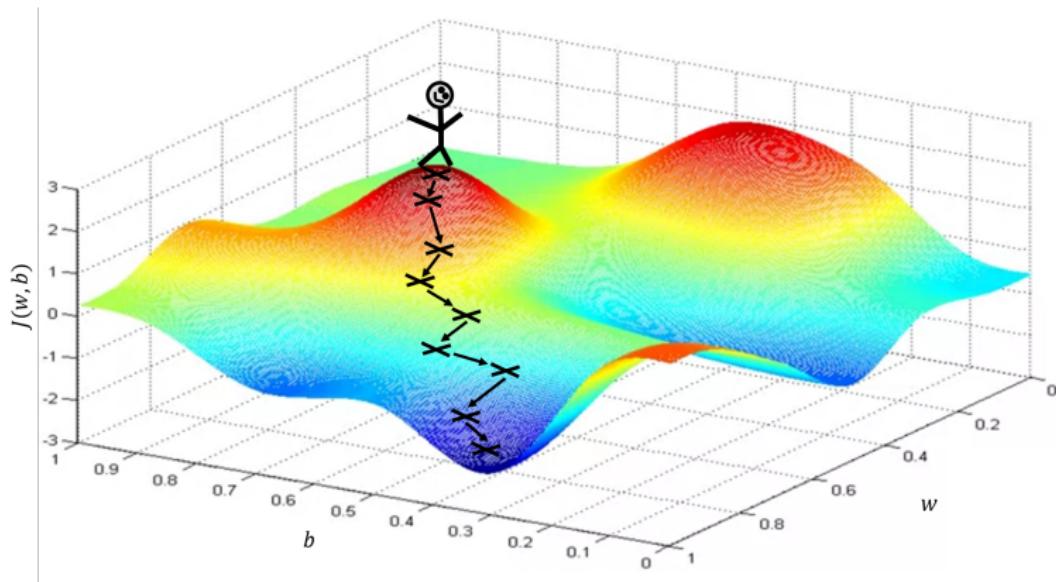
Após olhar ao redor, você decide avançar na direção indicada.



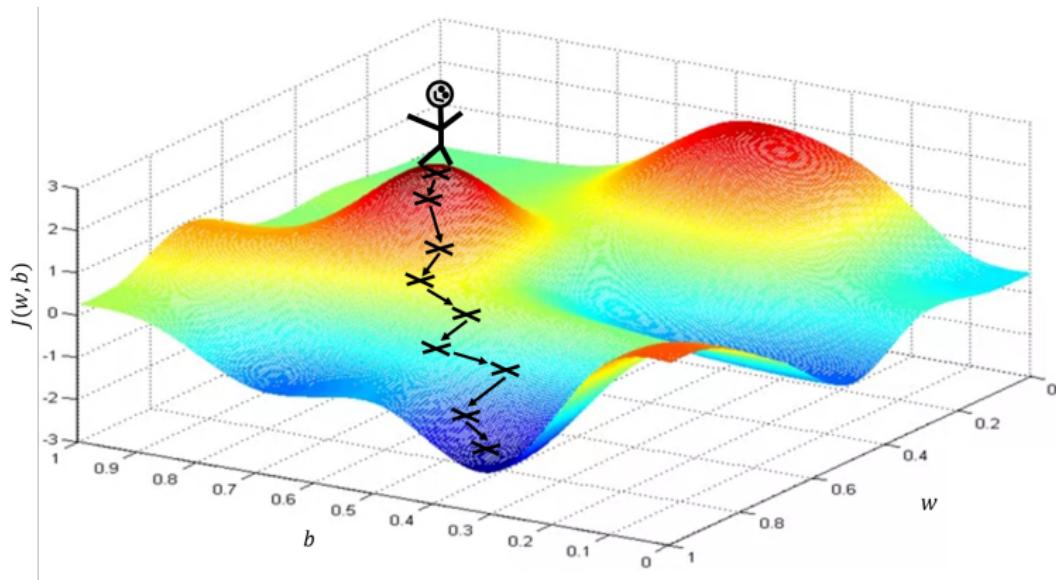
Agora você repete o processo. Você olha ao redor e decide seu próximo passo. Você então decide avançar mais um pouco.



Você pode repetir esse processo iterativamente até que você chegue no ponto desejado.



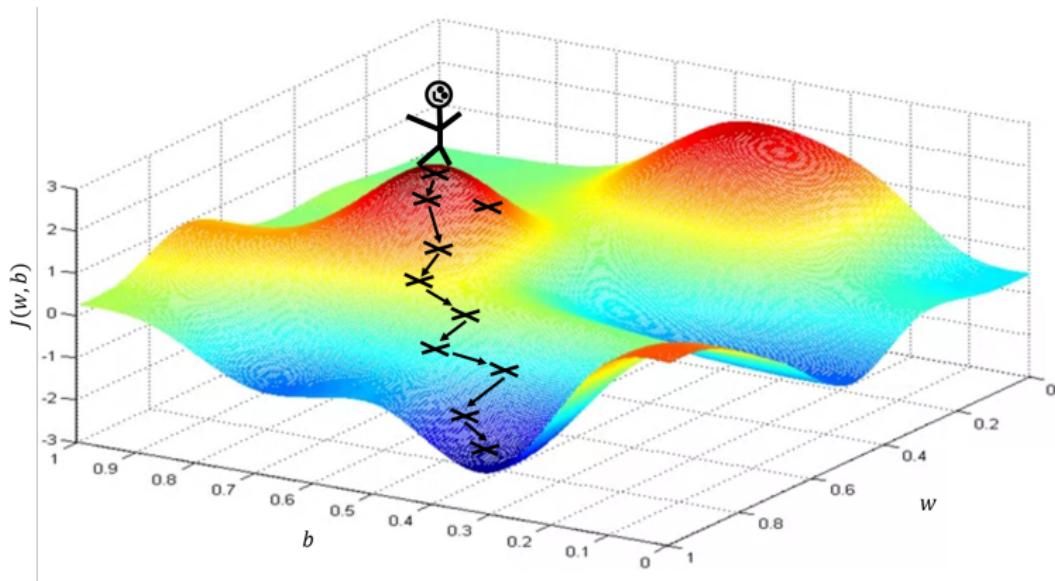
Você pode repetir esse processo iterativamente até que você chegue no ponto desejado.



Observação

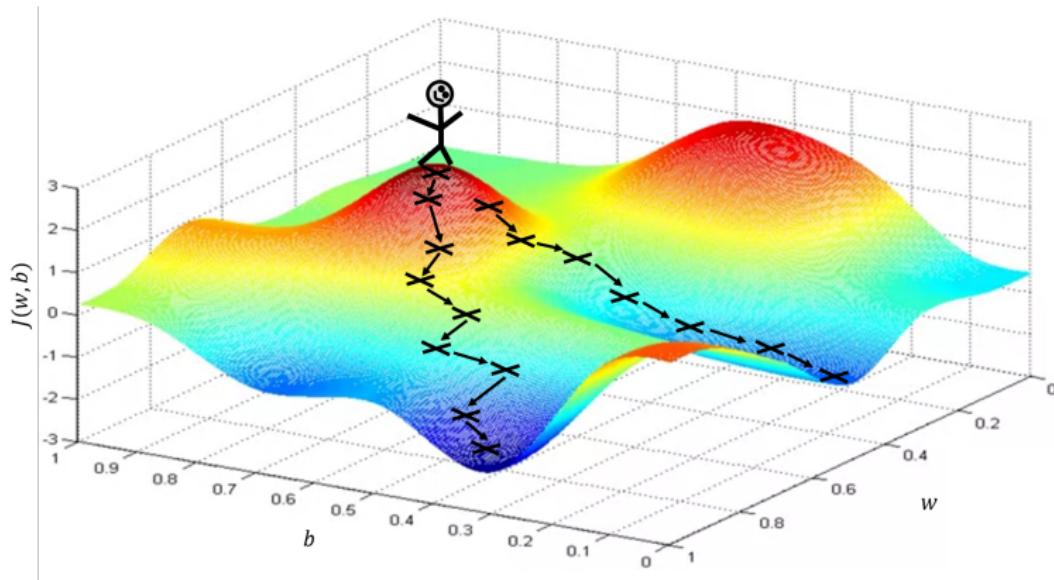
O Método do Gradiente faz justamente isso. Por isso, ele também é chamado de método do gradiente descendente, do inglês, *gradient descent*.

Lembre-se que você precisa escolher a posição inicial no Método do Gradiente. Vamos considerar agora que você escolheu uma posição inicial um pouco mais à direita.



Método do Gradiente

Acontece que, no Método do Gradiente, ainda que você escolha uma posição inicial apenas ligeiramente diferente, você pode acabar com parâmetros w e b bem diferentes...



Pergunta:

Qual solução é um ótimo local e qual é um ótimo global?

Pergunta:

Mas como implementamos na prática o Método do Gradiente?

Resposta:

Veremos isso agora

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

Método do Gradiente: Como implementar

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- "Atualize seu parâmetro w a partir do seu valor atual ajustado por um pequeno valor dado pelo termo $\alpha \frac{d}{dw} J(w, b)$ "

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- "Atualize seu parâmetro w a partir do seu valor atual ajustado por um pequeno valor dado pelo termo $\alpha \frac{d}{dw} J(w, b)$ "
- o sinal = na expressão denota a operação de atribuição (não é igualdade) → (exemplo parecido: $i = i + 1$)

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- "Atualize seu parâmetro w a partir do seu valor atual ajustado por um pequeno valor dado pelo termo $\alpha \frac{d}{dw} J(w, b)$ "
- o sinal = na expressão denota a operação de atribuição (não é igualdade) → (exemplo parecido: $i = i + 1$)
- símbolo α (letra grega "alfa") denota a chamada taxa de aprendizado → com ela, você consegue controlar o quanto você quer que o termo $\frac{d}{dw} J(w, b)$ impacte na atualização do parâmetro w

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- "Atualize seu parâmetro w a partir do seu valor atual ajustado por um pequeno valor dado pelo termo $\alpha \frac{d}{dw} J(w, b)$ "
- o sinal = na expressão denota a operação de atribuição (não é igualdade) → (exemplo parecido: $i = i + 1$)
- símbolo α (letra grega "alfa") denota a chamada taxa de aprendizado → com ela, você consegue controlar o quanto você quer que o termo $\frac{d}{dw} J(w, b)$ impacte na atualização do parâmetro w
- um valor elevado para α indica um processo de aprendizado agressivo, onde w será drasticamente atualizado pelo termo $\frac{d}{dw} J(w, b)$ (cuidado! você pode acabar se perdendo na sua busca pelo mínimo da função)

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- "Atualize seu parâmetro w a partir do seu valor atual ajustado por um pequeno valor dado pelo termo $\alpha \frac{d}{dw} J(w, b)$ "
- o sinal = na expressão denota a operação de atribuição (não é igualdade) → (exemplo parecido: $i = i + 1$)
- símbolo α (letra grega "alfa") denota a chamada taxa de aprendizado → com ela, você consegue controlar o quanto você quer que o termo $\frac{d}{dw} J(w, b)$ impacte na atualização do parâmetro w
- um valor elevado para α indica um processo de aprendizado agressivo, onde w será drasticamente atualizado pelo termo $\frac{d}{dw} J(w, b)$ (cuidado! você pode acabar se perdendo na sua busca pelo mínimo da função)
- um valor muito pequeno para α indica um processo de aprendizado lento, onde w será pouco afetado pelo termo $\frac{d}{dw} J(w, b)$

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- "Atualize seu parâmetro w a partir do seu valor atual ajustado por um pequeno valor dado pelo termo $\alpha \frac{d}{dw} J(w, b)$ "
- o sinal = na expressão denota a operação de atribuição (não é igualdade) → (exemplo parecido: $i = i + 1$)
- símbolo α (letra grega "alfa") denota a chamada taxa de aprendizado → com ela, você consegue controlar o quanto você quer que o termo $\frac{d}{dw} J(w, b)$ impacte na atualização do parâmetro w
- um valor elevado para α indica um processo de aprendizado agressivo, onde w será drasticamente atualizado pelo termo $\frac{d}{dw} J(w, b)$ (cuidado! você pode acabar se perdendo na sua busca pelo mínimo da função)
- um valor muito pequeno para α indica um processo de aprendizado lento, onde w será pouco afetado pelo termo $\frac{d}{dw} J(w, b)$
- $\frac{d}{dw} J(w, b)$ é a derivada da função $J(w, b)$ em relação ao parâmetro w → mostra a direção em que $J(w, b)$ mais cresce para uma pequena variação de w .

Método do Gradiente: Como implementar

Em cada passo (iteração), devemos...

Atualizar w da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- "Atualize seu parâmetro w a partir do seu valor atual ajustado por um pequeno valor dado pelo termo $\alpha \frac{d}{dw} J(w, b)$ "
- o sinal = na expressão denota a operação de atribuição (não é igualdade) → (exemplo parecido: $i = i + 1$)
- símbolo α (letra grega "alfa") denota a chamada taxa de aprendizado → com ela, você consegue controlar o quanto você quer que o termo $\frac{d}{dw} J(w, b)$ impacte na atualização do parâmetro w
- um valor elevado para α indica um processo de aprendizado agressivo, onde w será drasticamente atualizado pelo termo $\frac{d}{dw} J(w, b)$ (cuidado! você pode acabar se perdendo na sua busca pelo mínimo da função)
- um valor muito pequeno para α indica um processo de aprendizado lento, onde w será pouco afetado pelo termo $\frac{d}{dw} J(w, b)$
- $\frac{d}{dw} J(w, b)$ é a derivada da função $J(w, b)$ em relação ao parâmetro w → mostra a direção em que $J(w, b)$ mais cresce para uma pequena variação de w .

Observação

Lembre-se que também temos o parâmetro b no nosso modelo, então também teremos que atualizá-lo a partir da expressão equivalente

$$b = b - \alpha \frac{d}{db} J(w, b)$$

Em cada passo (iteração), devemos...

Atualizar w e b da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

$$b = b - \alpha \frac{d}{db} J(w, b)$$

Importante:

- Devemos repetir esse processo até a **convergência**, ou seja, até que w e b deixem de atualizar (isso ocorre quando estamos próximos de um mínimo da função J)

Em cada passo (iteração), devemos...

Atualizar w e b da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

$$b = b - \alpha \frac{d}{db} J(w, b)$$

Forma **correta** de implementação (atualização simultânea):

$$tmp_w = w - \alpha \frac{d}{dw} J(w, b)$$

$$tmp_b = b - \alpha \frac{d}{db} J(w, b)$$

$$w = tmp_w$$

$$b = tmp_b$$

Em cada passo (iteração), devemos...

Atualizar w e b da seguinte maneira:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

$$b = b - \alpha \frac{d}{db} J(w, b)$$

Forma **INCORRETA** de implementação:

$$\text{tmp_}w = w - \alpha \frac{d}{dw} J(w, b)$$

$$w = \text{tmp_}w$$

$$\text{tmp_}b = b - \alpha \frac{d}{db} J(w, b)$$

$$b = \text{tmp_}b$$

Pergunta:

O Método do Gradiente é um algoritmo que visa encontrar os valores dos parâmetros w e b que minimizam a função custo J . O que a expressão abaixo faz? (assuma que α é pequeno)

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

- A) Checa-se w é igual a $w - \alpha \frac{d}{dw} J(w, b)$
- B) Atualiza w ligeiramente

Vimos até agora que o Método do Gradiente pode ser implementado da seguinte forma:

repetir até a convergência:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

$$b = b - \alpha \frac{d}{db} J(w, b)$$

onde α denota a taxa de aprendizado (geralmente um valor positivo $\alpha > 0$ pequeno).

Pergunta:

E se quisermos encontrar o mínimo de uma função f que possui apenas 1 parâmetro? Por exemplo, $J(w)$?

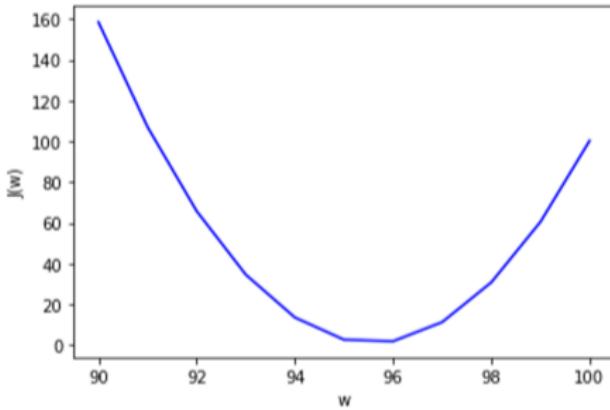
Resposta:

Nesse caso, precisamos fazer apenas

$$w = w - \alpha \frac{d}{dw} J(w)$$

Exemplo

Na atividade de programação da última aula, obtivemos a seguinte função $J(w)$ quando consideramos $b = 0$:



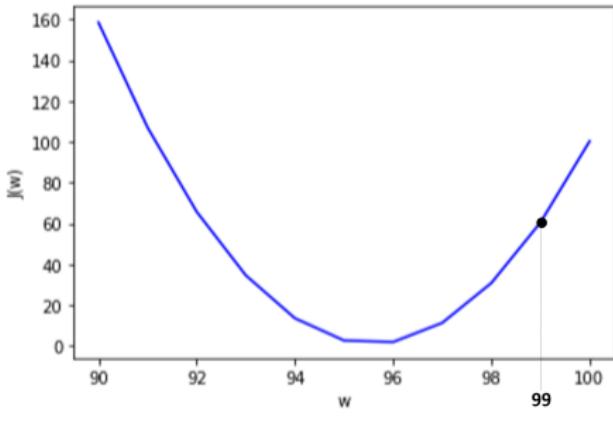
Pergunta:

Como ficaria a aplicação do Método do Gradiente para esse caso? Ou seja, como ficaria a atualização

$$w = w - \alpha \frac{d}{dw} J(w) ?$$

Exemplo

Considerando uma inicialização em $w = 99$, temos:

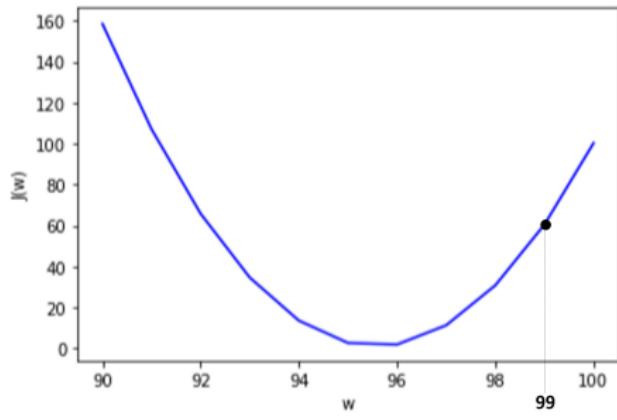


Pergunta:

Sabendo que a derivada é a inclinação da reta que tangencia o ponto $w = 99$, teremos $\frac{d}{dw} J(w) > 0$ ou $\frac{d}{dw} J(w) < 0$?

Exemplo

Considerando uma inicialização em $w = 99$, temos:



Observação:

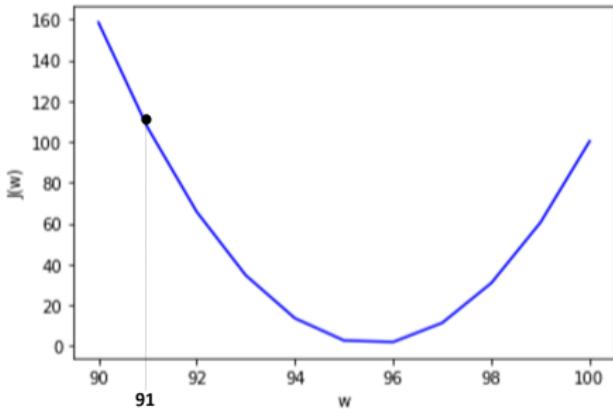
Como $\frac{d}{dw} J(w) > 0$ para $w = 99$, note que teremos

$$w = w - \alpha \text{ (valor positivo)}$$

Ou seja, o valor atualizado para w será **menor** que 99, já que $\alpha > 0$. Para um valor apropriado para α , estaremos caminhando em direção ao mínimo da função $J(w)$.

Exemplo

Supondo agora uma inicialização em $w = 91$, temos:

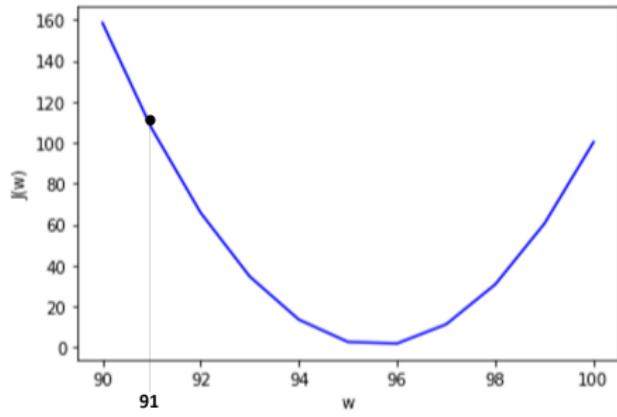


Pergunta:

Sabendo que a derivada é a inclinação da reta que tangencia o ponto $w = 91$, teremos $\frac{d}{dw} J(w) > 0$ ou $\frac{d}{dw} J(w) < 0$?

Exemplo

Considerando uma inicialização em $w = 91$, temos:

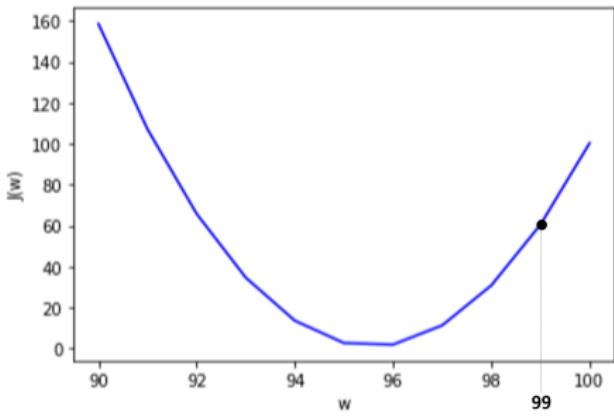


Observação:

Como $\frac{d}{dw} J(w) < 0$ para $w = 91$, note que teremos

$$w = w - \alpha (\text{valor negativo})$$

Ou seja, o valor atualizado para w será **maior** que 91, já que $\alpha > 0$. Para um valor apropriado para α , estaremos caminhando em direção ao mínimo da função $J(w)$.



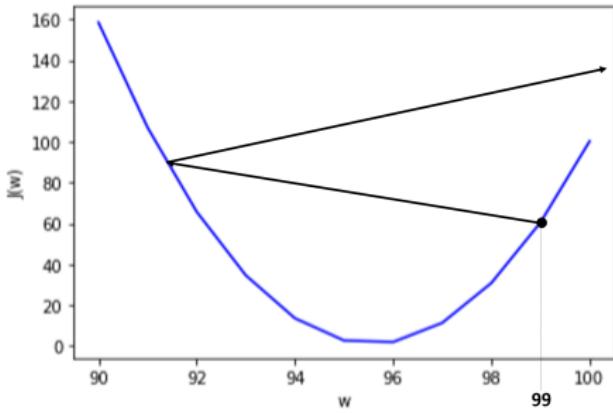
Pergunta:

O que acontece se tivermos um valor excessivamente grande para a taxa de aprendizado α , por exemplo, $\alpha = 1000000$?

Apenas relembrando que

$$w = w - \alpha \frac{d}{dw} J(w)$$

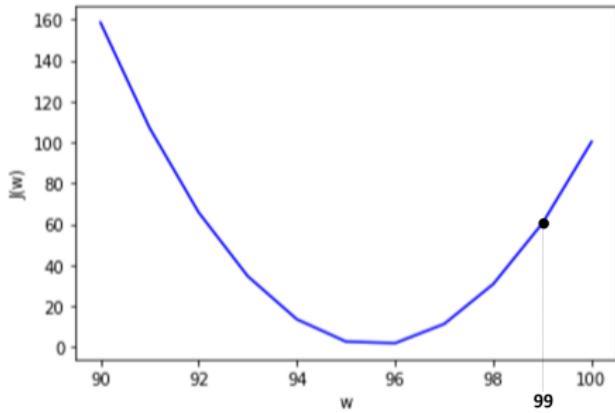
Exemplo



Resposta:

O Método do Gradiente irá divergir, e o valor de w que minimiza $J(w)$ nunca será encontrado.

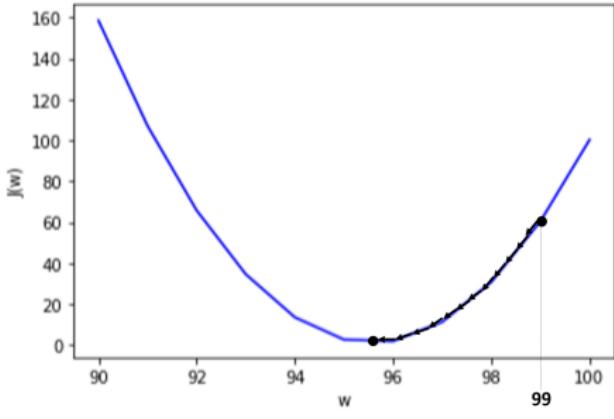
Exemplo



Pergunta:

E se tivermos um valor excessivamente pequeno para a taxa de aprendizado α , por exemplo, $\alpha = 0.000001$?

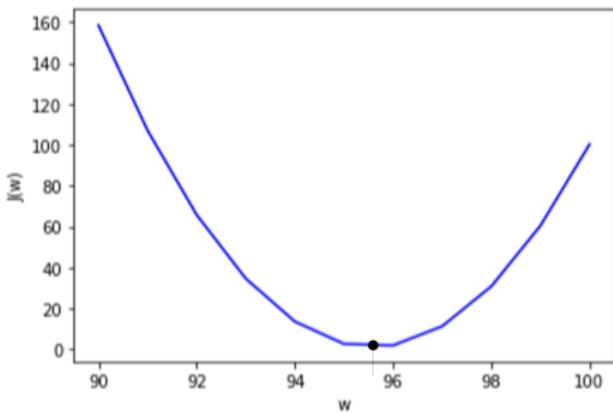
Exemplo



Resposta:

O Método do Gradiente será lento, e demorará muitas iterações até convergir para o mínimo.

Exemplo



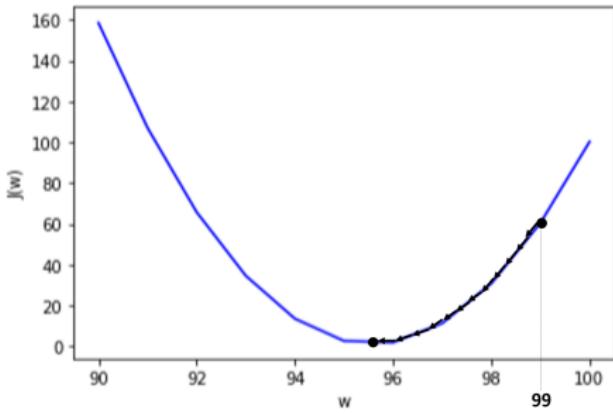
Pergunta:

O que acontece se inicializarmos w já no ponto de mínimo da função $J(w)$?

Resposta:

O método permanecerá no mínimo, já que, no mínimo, temos $\frac{d}{dw} J(w) = 0$.

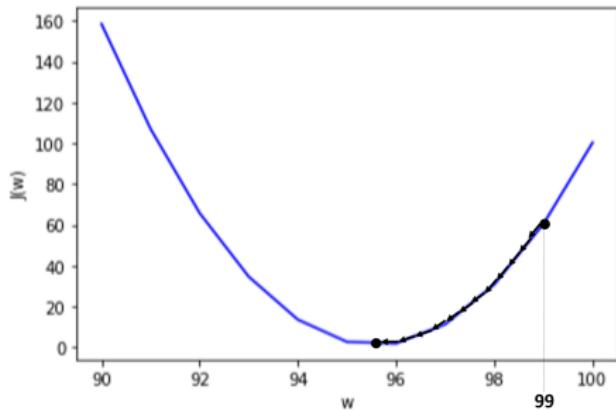
Exemplo



Observação:

Note que $\frac{d}{dw} J(w)$ diminui (em termos do seu módulo) à medida com que nos aproximamos do mínimo da função. Isso significa que, utilizando um valor pequeno e fixo para α , passos cada vez menores em direção ao mínimo são dados ao longo das iterações.

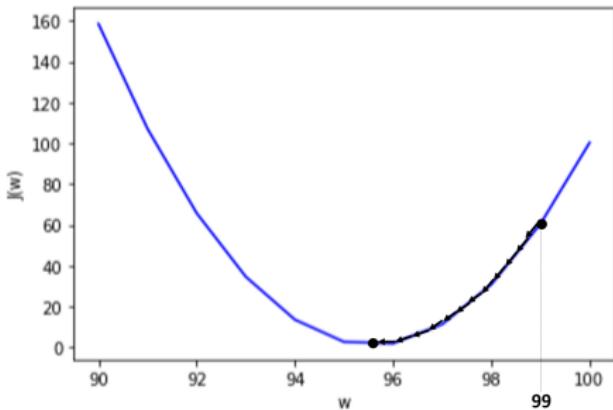
Exemplo



Pergunta:

Como fazemos então para selecionarmos um valor adequado para α ?

Exemplo



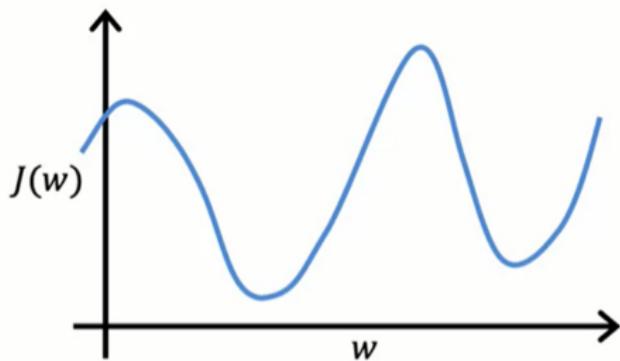
Pergunta:

Como fazemos então para selecionarmos um valor adequado para α ?

Resposta:

Depende muito de cada problema. Dica: teste diferentes valores!

Exemplo



Pergunta:

Caso estejamos minimizando uma função $J(w)$ que possui diversos mínimos locais, é possível que o Método do Gradiente não converja para o mínimo global?

Resposta:

Sim. Note que o mínimo para o qual o método irá convergir depende do ponto de inicialização.

Voltando para o nosso problema de regressão linear

Queremos utilizar o **Método do Gradiente** para encontrar os parâmetros w, b do modelo $f_{w,b}(x) = wx + b$ que minimizam a função custo

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

Em outras palavras, queremos usar o Método do Gradiente para encontrar a **reta que melhor representa nossos dados**.

Para fazermos isso, basta escolhermos valores iniciais para w, b e...

repetir até a convergência:

$$w = w - \alpha \frac{d}{dw} J(w, b)$$

$$b = b - \alpha \frac{d}{db} J(w, b)$$

Para essa função custo, é possível mostrar que

$$\frac{d}{dw} J(w, b) = \frac{1}{m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

$$\frac{d}{db} J(w, b) = \frac{1}{m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)$$

Como deduzir as expressões acima?

Ou seja, encontraremos os parâmetros w, b do modelo $f_{w,b}(x) = wx + b$ que minimizam a função custo

$$J(w, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)^2$$

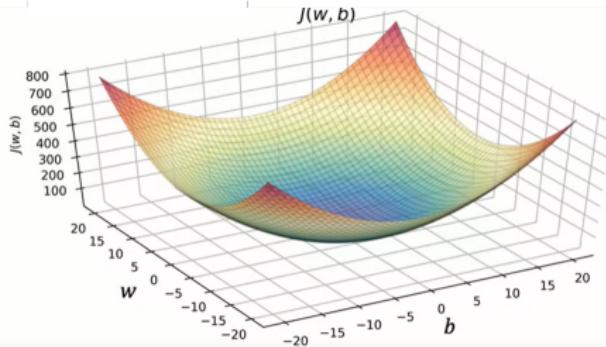
escolhendo valores iniciais para w, b e então repetindo até a convergência as seguintes atualizações:

$$w = w - \alpha \frac{1}{m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^m \left(f_{w,b}(x^{(i)}) - y^{(i)} \right)$$

Observação importante:

Em problemas de **regressão linear**, a função quadrática J é **convexa**, ou seja, ela não possui outros mínimos locais além do próprio mínimo global.



Conhecendo esse resultado teórico, sabemos que o Método do Gradiente nos levará, obrigatoriamente, para o mínimo global da função.

De olho no código!

Vamos agora ver como implementar na prática o **Método do Gradiente**

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximppv2/master/blob/main/codigo_aula5_metodo_do_gradiente.ipynb

Parte 1

Rode todo o código. Responda às questões nele contidas e complete-o, se necessário.

Parte 2

Insira no código da Parte 1 o conjunto de medições que você já criou anteriormente para um resistor de 50Ω , faça as adaptações necessárias e verifique os resultados.

Regressão Linear Múltipla (Parte 1)



Onde estamos e para onde vamos?

Nas aula anteriores, aprendemos a fazer regressão linear com uma única variável:

Área da casa [m ²] (x)	Custo em R\$ (y)
32	51.000
149	265.000
78	110.000
...	...
220	315.000

Variável 1 (característica 1): Área da casa

Modelo Utilizado:

$$f_{w,b} = wx + b$$

Onde estamos e para onde vamos?

Agora estudaremos regressão linear com **múltiplas variáveis**:

Área da casa [m ²] (x_1)	Número de quartos (x_2)	Andares (x_3)	Idade [anos] (x_4)	Custo em R\$ (y)
32	1	1	2	51.000
149	3	2	10	265.000
78	2	1	30	110.000
...
220	4	2	5	315.000

Variável 1 (característica 1): Área da casa

Variável 2 (característica 2): Número de quartos

Variável 3 (característica 3): Número de andares

Variável 4 (característica 4): idade da construção

Levando em conta mais características, é esperado que o modelo seja capaz de prever melhor o valor de uma casa?

Regressão Linear Múltipla

Regressão Linear Múltipla

Área da casa [m ²] (x_1)	Número de quartos (x_2)	Andares (x_3)	Idade [anos] (x_4)	Custo em R\$ (y)
32	1	1	2	51.000
149	3	2	10	265.000
78	2	1	30	110.000
...
220	4	2	5	315.000

Notação:

$$x_j = j\text{-ésima característica} \rightarrow j = 1, 2, \dots, 4$$

$$n = \text{número total de características} \rightarrow n = 4$$

$$\vec{x}^{(i)} = \text{características do } i\text{-ésimo exemplo de treinamento} \rightarrow \vec{x}^{(2)} = [\begin{array}{cccc} 149 & 3 & 2 & 10 \end{array}]$$

$$\vec{x}_j^{(i)} = \text{valor da característica } j \text{ do } i\text{-ésimo exemplo de treinamento} \rightarrow \vec{x}_2^{(2)} = 3$$

Observação:

Para simplificar a notação, vamos tratar a sobre-barra como um elemento opcional de notação, tal que $\vec{x}_j^{(2)} = x_j^{(2)}$, por exemplo. \rightarrow (serve apenas para enfatizar que trata-se de um vetor).

Pergunta

Seja o conjunto de dados abaixo. Quanto vale $\vec{x}_4^{(3)}$?

Área da casa [m ²] (x_1)	Número de quartos (x_2)	Andares (x_3)	Idade [anos] (x_4)	Custo em R\$ (y)
32	1	1	2	51.000
149	3	2	10	265.000
78	2	1	30	110.000
...
220	4	2	5	315.000

Regressão Linear Múltipla

Antes, na regressão linear com uma única variável, tínhamos o seguinte modelo:

$$f_{w,b}(x) = wx + b$$

Agora, na regressão linear com múltiplas variáveis, teremos:

$$f_{w,b}(x) = w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b$$

Exemplo:

$$f_{w,b}(x) = 0.1x_1 + 4x_2 + 10x_3 - 2x_4 + 80 \quad \rightarrow \quad \text{valor da casa em milhares de R\$}$$

x_1 : Área da casa

x_2 : Número de quartos

x_3 : Número de andares

x_4 : idade da construção

Olhando com detalhes o modelo

Área da casa [m ²] (x_1)	Número de quartos (x_2)	Andares (x_3)	Idade [anos] (x_4)	Custo em R\$ (y)
32	1	1	2	51.000
149	3	2	10	265.000
78	2	1	30	110.000
...
220	4	2	5	315.000

O que significam os coeficientes do modelo?

$$f_{w,b}(x) = 0.1x_1 + 4x_2 + 10x_3 - 2x_4 + 80 \quad \rightarrow \quad \text{valor da casa em milhares de R\$}$$

- Note que cada m² adiciona R\$ 100 ao valor da casa
- Cada quarto adiciona R\$ 4000 ao valor da casa
- Cada andar adiciona R\$ 10000 ao valor da casa
- Cada ano **reduz** em R\$ 2000 o valor da casa
- R\$ 80000 seria o “valor base” de qualquer casa

Olhando com detalhes o modelo

Área da casa [m ²] (x_1)	Número de quartos (x_2)	Andares (x_3)	Idade [anos] (x_4)	Custo em R\$ (y)
32	1	1	2	51.000
149	3	2	10	265.000
78	2	1	30	110.000
...
220	4	2	5	315.000

O modelo abaixo é bom para estimar o preço das casas?

$$f_{w,b}(x) = 0.1x_1 + 4x_2 + 10x_3 - 2x_4 + 80 \quad \rightarrow \quad \text{valor da casa em milhares de R\$}$$

- Para a primeira casa do conjunto de dados, temos:

$$f_{w,b}(x) = 0.1 \cdot 32 + 4 \cdot 1 + 10 \cdot 1 - 2 \cdot 2 + 80 = 93.2$$

- Para as demais casas, temos

$$f_{w,b}(x) = 0.1 \cdot 149 + 4 \cdot 3 + 10 \cdot 2 - 2 \cdot 10 + 80 = 106.9$$

$$f_{w,b}(x) = 0.1 \cdot 78 + 4 \cdot 2 + 10 \cdot 1 - 2 \cdot 30 + 80 = 45.8$$

$$f_{w,b}(x) = 0.1 \cdot 220 + 4 \cdot 4 + 10 \cdot 2 - 2 \cdot 5 + 80 = 128.0$$

Olhando com detalhes o modelo

Área da casa [m ²] (x_1)	Número de quartos (x_2)	Andares (x_3)	Idade [anos] (x_4)	Custo em R\$ (y)
32	1	1	2	51.000
149	3	2	10	265.000
78	2	1	30	110.000
...
220	4	2	5	315.000

Conclusão

Observando como o modelo em tela se comporta para os dados que temos, parece que um modelo mais assertivo poderia ter sido obtido.

Pergunta:

Como obter um modelo mais preciso?

Um modelo com n características

Um modelo com n características é dado por

$$f_{w,b}(x) = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

Parâmetros do modelo:

$$\vec{w} = [\quad w_1 \quad w_2 \quad \cdots \quad w_n \quad]$$

b \rightarrow b não é um vetor, mas sim um escalar!

Características presentes no modelo:

$$\vec{x} = [\quad x_1 \quad x_2 \quad \cdots \quad x_n \quad]$$

Utilizando essa notação, note que podemos reescrever $f_{w,b}(x)$ na seguinte forma compacta:

$$f_{w,b}(x) = \vec{w} \cdot \vec{x} + b$$

onde \cdot denota o produto escalar, tal que

$$\vec{w} \cdot \vec{x} = w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

Notação sem vetorização

Modelo:

$$f_{w_1, w_2, \dots, w_n, b}(x_1, x_2, \dots, x_n) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + b$$

Parâmetros:

$$w_1, w_2, \dots, w_n \text{ e } b$$

Função custo:

$$J(w_1, w_2, \dots, w_n, b)$$

Método do Gradiente consiste em repetir até convergir:

$$w_j = w_j - \alpha \frac{d}{dw_j} J(w_1, w_2, \dots, w_n, b)$$

$$b = b - \alpha \frac{d}{db} J(w_1, w_2, \dots, w_n, b)$$

Notação com vetorização

Modelo:

$$f_{\vec{w}, b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

Parâmetros:

$$\vec{w} \quad \text{e} \quad b$$

Função custo:

$$J(\vec{w}, b)$$

Método do Gradiente consiste em repetir até convergir:

$$w_j = w_j - \alpha \frac{d}{dw_j} J(\vec{w}, b)$$

$$b = b - \alpha \frac{d}{db} J(\vec{w}, b)$$

Perguntas:

- Agora que temos n características, quanto vale $\frac{d}{dw_j} J(\vec{w}, b)$ para cada w_j ?
- Agora que temos n características, quanto vale $\frac{d}{db} J(\vec{w}, b)$?

É possível mostrar que

$$\frac{d}{dw_j} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right) \vec{x}_j^{(i)}$$

$$\frac{d}{db} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right)$$

Portanto, o Método do Gradiente aplicado ao contexto de Regressão Linear múltipla consiste em...

Repetir até convergir:

$$w_1 = w_1 - \alpha \frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right) \vec{x}_1^{(i)}$$

⋮

$$w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right) \vec{x}_n^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right)$$

De olho no código!

Vamos agora ver como implementar na prática o **Método do Gradiente para Regressão linear Múltipla**

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula6_regressao_linear_multipla.ipynb

Parte 1

Rode todo o código. Responda às questões nele contidas e complete-o, se necessário.

Parte 2

- 1** Quais foram os valores obtidos para \vec{w} e b ?
- 2** Esses valores são os melhores possíveis?
- 3** O que pode estar acontecendo?
- 4** Rode novamente o Método do Gradiente inicializando \vec{w} e b num local mais próximo do ótimo. Quais foram os valores agora obtidos para \vec{w} e b ?

Tópico adicional: Vetorização

Esta aula constitui um tópico adicional da disciplina. Trata-se de um conteúdo opcional. Sua atividade não valerá nota e não precisa ser enviada.



Ao implementar um algoritmo de Aprendizado de Máquina, a vetorização provê dois benefícios principais:

- 1 Seu código torna-se mais compacto.
- 2 Seu algoritmo é capaz de rodar de forma mais rápida e eficiente.

Um código escrito de forma vetorizada é também capaz de extrair máximo proveito de:

- bibliotecas numéricas modernas e otimizadas para cálculo matemático, como a biblioteca de Álgebra Linear **NumPy**, por exemplo.
- elementos de hardware voltados ao processamento intenso de dados, como GPUs (*Graphical Processing Units*), por exemplo.

Exemplo de Vetorização

Sejam os seguintes parâmetros e características de um modelo:

$$\vec{w} = [\ w_1 \quad w_2 \quad w_3 \] \rightarrow (n = 3)$$

b (escalar)

$$\vec{x} = [\ x_1 \quad x_2 \quad x_3 \]$$

Em Álgebra Linear, a indexação (contagem de elementos) começa em 1.

Em código escrito em Python (NumPy), a indexação começa em 0.

Exemplo:

```
w = np.array([1.0, 2.5, -3.3])
```

```
b = 4
```

```
x = np.array([10, 20, 30])
```

Para acessar o primeiro elemento de w , usamos $w[0]$.

Para acessar o segundo elemento de w , usamos $w[1]$.

Para acessarmos o terceiro elemento de w , usamos $w[2]$.

Idem para os elementos do vetor x .

Exemplo de Vetorização

Suponha que você quer implementar uma previsão feita pelo modelo

$$f_{\vec{w}, b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

Código sem vetorização:

```
f = w[0]*x[0] + w[1]*x[1] + w[2]*x[2] + b
```

Pergunta:

Seria fácil programar a linha de código acima se tivéssemos $n = 1000$?

Exemplo de Vetorização

Código ainda sem vetorização, mas usando **loop for**:

Sabemos que

$$f_{\vec{w}, b}(\vec{x}) = \left(\sum_{j=1}^n w_j x_j \right) + b$$

Portanto poderíamos usar o seguinte código:

```
f = 0  
for j in range(0,n):  
    f = f + w[j] * x[j]  
f = f+b
```

Observação 1: Em Python, `j in range(0,n)` significa que j será $0, 1, 2, \dots, n - 1$.

Observação 2: Em Python, o comando `range(0,n)` faz a mesma coisa que `range(n)`

Pergunta:

Apesar de ser melhor que a nossa primeira implementação, essa segunda seria ainda a forma mais otimizada para a realização dos cálculos?

Código COM vetorização:

Sabendo que

$$f_{\vec{w}, b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

podemos implementar tal operação matemática usando uma **única linha de código**:

```
f = np.dot(w,x)+b
```

Exemplo de Vetorização

Observação:

Ao utilizarmos comandos da biblioteca NumPy, como `f = np.dot(w,x)+b`, por exemplo, estamos otimizando o nosso código, especialmente para os casos em que n é grande.

Isso acontece pois, por trás, a biblioteca **NumPy** utiliza **paralelismo de hardware** para a realização das operações matemáticas, mesmo que você esteja usando uma CPU comum ao invés de uma GPU.

Um segundo exemplo de Vetorização

Suponha que você deseja implementar o Método do Gradiente para um modelo com $b = 0$, $\vec{w} = (w_1, w_2, \dots, w_{16})$, derivadas $\vec{d} = (d_1, d_2, \dots, d_{16})$ e $\alpha = 0.1$, tal que

$$w_j = w_j - 0.1d_j$$

Código sem vetorização:

```
for j in range(0,16):
    w[j] = w[j] - 0.1*d[j]
```

Código COM vetorização:

```
w = w - 0.1*d
```

De olho no código!

De olho no código!

Vamos agora aprender conceitos mais específicos acerca de Python, NumPy e Vetorização

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximppv2/master/blob/main/codigo_aula6_topico_adicional.ipynb

Parte 1

Rode todo o “codigo - Python, NumPy e Vetorização.ipynb” sem fazer qualquer tipo de alteração. Certifique-se de que você o compreendeu.

Parte 2

- 1 Qual foi a diferença de tempo observada entre rodar comandos usando “loop for” versus vetorização?

Escalonamento de características e escolha do α



Nas aulas anteriores, mostramos como o Método do Gradiente pode ser usado para resolver diversos tipos de problemas

Nesta aula, aprenderemos como aprimorar esse método por meio do **Escalonamento de Características** e de uma escolha apropriada para a **taxa de aprendizado α** .

Escalonamento de Características

Vamos começar com a técnica de **Escalonamento de Características**. Tal técnica faz com que o Método do Gradiente torne-se significativamente mais rápido.

Relação entre características e valores dos parâmetros

Suponha que você deseja obter um modelo que seja capaz de estimar o preço de uma casa:

$$\widehat{\text{preço}} = w_1 x_1 + w_2 x_2 + b$$

onde

- x_1 denota o tamanho em **feet²** (um valor tipicamente entre 300 e 2000)
- x_2 denota o número de quartos (um valor tipicamente entre 0 e 5)

Relação entre características e valores dos parâmetros

Suponha que você deseja obter um modelo que seja capaz de estimar o preço de uma casa:

$$\widehat{\text{preço}} = w_1 x_1 + w_2 x_2 + b$$

onde

- x_1 denota o tamanho em **feet²** (um valor tipicamente entre 300 e 2000)
- x_2 denota o número de quartos (um valor tipicamente entre 0 e 5)

Observações:

- Para que ambos os termos $w_1 x_1$ e $w_2 x_2$ tenham impacto significativo no cálculo do preço da casa, note que w_1 tenderá a ser pequeno em comparação com w_2 , já que x_1 é um valor tipicamente maior que x_2 .

Relação entre características e valores dos parâmetros

Suponha que você deseja obter um modelo que seja capaz de estimar o preço de uma casa:

$$\widehat{\text{preço}} = w_1 x_1 + w_2 x_2 + b$$

onde

- x_1 denota o tamanho em **feet²** (um valor tipicamente entre 300 e 2000)
- x_2 denota o número de quartos (um valor tipicamente entre 0 e 5)

Observações:

- Para que ambos os termos $w_1 x_1$ e $w_2 x_2$ tenham impacto significativo no cálculo do preço da casa, note que w_1 tenderá a ser pequeno em comparação com w_2 , já que x_1 é um valor tipicamente maior que x_2 .
- Isso significa que, enquanto o Método do Gradiente estiver buscando valores apropriados para w_1 e w_2 , o valor da função custo J será **muito mais sensível** a um incremento unitário em w_1 em comparação com um incremento unitário em w_2 .

Relação entre características e valores dos parâmetros

Suponha que você deseja obter um modelo que seja capaz de estimar o preço de uma casa:

$$\widehat{\text{preço}} = w_1x_1 + w_2x_2 + b$$

onde

- x_1 denota o tamanho em **feet²** (um valor tipicamente entre 300 e 2000)
- x_2 denota o número de quartos (um valor tipicamente entre 0 e 5)

Observações:

- Para que ambos os termos w_1x_1 e w_2x_2 tenham impacto significativo no cálculo do preço da casa, note que w_1 tenderá a ser pequeno em comparação com w_2 , já que x_1 é um valor tipicamente maior que x_2 .
- Isso significa que, enquanto o Método do Gradiente estiver buscando valores apropriados para w_1 e w_2 , o valor da função custo J será **muito mais sensível** a um incremento unitário em w_1 em comparação com um incremento unitário em w_2 .
- Se o Método do Gradiente “erra” um pouco na escolha de w_1 , a performance do modelo deteriora consideravelmente.

Relação entre características e valores dos parâmetros

Suponha que você deseja obter um modelo que seja capaz de estimar o preço de uma casa:

$$\widehat{\text{preço}} = w_1x_1 + w_2x_2 + b$$

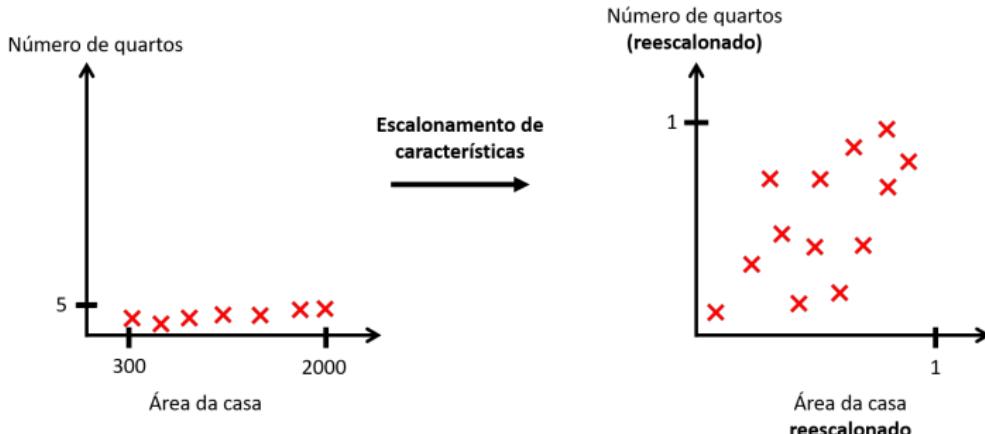
onde

- x_1 denota o tamanho em **feet²** (um valor tipicamente entre 300 e 2000)
- x_2 denota o número de quartos (um valor tipicamente entre 0 e 5)

Observações:

- Para que ambos os termos w_1x_1 e w_2x_2 tenham impacto significativo no cálculo do preço da casa, note que w_1 tenderá a ser pequeno em comparação com w_2 , já que x_1 é um valor tipicamente maior que x_2 .
- Isso significa que, enquanto o Método do Gradiente estiver buscando valores apropriados para w_1 e w_2 , o valor da função custo J será **muito mais sensível** a um incremento unitário em w_1 em comparação com um incremento unitário em w_2 .
- Se o Método do Gradiente “erra” um pouco na escolha de w_1 , a performance do modelo deteriora consideravelmente.
- Por outro lado, se escalonarmos os dados antes de aplicar o método do gradiente, criando duas **características escalonadas** entre 0 e 1 (por exemplo), o Método do Gradiente torna-se mais rápido para encontrar valores apropriados para w_1 e w_2 .

Escalonamento de Características



Observação

O escalonamento das características desempenha um papel fundamental no aumento de velocidade de convergência do método do gradiente, especialmente quando as características do problema possuem valores com ordens de grandeza diversas.

OPÇÃO 1: Dividindo pelo máximo

Se $300 \leq x_1 \leq 2000$, podemos escalar x_1 da seguinte maneira:

$$x_{1, \text{escalonado}} = \frac{x_1}{2000}$$

Assim, teremos $0.15 \leq x_{1, \text{escalonado}} \leq 1$

Similarmente, se $0 \leq x_2 \leq 5$, podemos escalar x_2 da seguinte maneira:

$$x_{2, \text{escalonado}} = \frac{x_2}{5}$$

Assim, teremos $0 \leq x_{2, \text{escalonado}} \leq 1$

OPÇÃO 2: Normalização pela média

Se $300 \leq x_1 \leq 2000$, podemos escalar x_1 da seguinte maneira:

$$x_{1,\text{escalonado}} = \frac{x_1 - \mu_1}{2000 - 300} \rightarrow \text{onde } \mu_1 \text{ é a média de } x_1$$

Supondo $\mu_1 = 600 \text{ feet}^2$, teremos $-0.18 \leq x_{1,\text{escalonado}} \leq 0.82$

Similarmente, se $0 \leq x_2 \leq 5$, podemos escalar x_2 da seguinte maneira:

$$x_{2,\text{escalonado}} = \frac{x_2 - \mu_2}{5 - 0} \rightarrow \text{onde } \mu_2 \text{ é a média de } x_2$$

Supondo $\mu_2 = 2.3$ quartos, teremos $-0.46 \leq x_{2,\text{escalonado}} \leq 0.54$

OPÇÃO 2: Normalização pela média

Se $300 \leq x_1 \leq 2000$, podemos escalar x_1 da seguinte maneira:

$$x_{1,\text{escalonado}} = \frac{x_1 - \mu_1}{2000 - 300} \rightarrow \text{onde } \mu_1 \text{ é a média de } x_1$$

Supondo $\mu_1 = 600 \text{ feet}^2$, teremos $-0.18 \leq x_{1,\text{escalonado}} \leq 0.82$

Similarmente, se $0 \leq x_2 \leq 5$, podemos escalar x_2 da seguinte maneira:

$$x_{2,\text{escalonado}} = \frac{x_2 - \mu_2}{5 - 0} \rightarrow \text{onde } \mu_2 \text{ é a média de } x_2$$

Supondo $\mu_2 = 2.3$ quartos, teremos $-0.46 \leq x_{2,\text{escalonado}} \leq 0.54$

Observação:

- Ao subtrair a média de uma sequência de números, a sequência resultante acaba ficando com média 0.

Exemplos de escalonamento

OPÇÃO 3: Normalização Z-score → Também chamada de **Padronização**

Se $300 \leq x_1 \leq 2000$, podemos escalar x_1 da seguinte maneira:

$$x_{1,\text{escalonado}} = \frac{x_1 - \mu_1}{\sigma_1} \rightarrow \text{onde } \sigma_1 \text{ é o desvio padrão de } x_1$$

Supondo $\mu_1 = 600 \text{ feet}^2$ e $\sigma_1 = 450$, teremos $-0.67 \leq x_{1,\text{escalonado}} \leq 3.1$

Similarmente, se $0 \leq x_2 \leq 5$, podemos escalar x_2 da seguinte maneira:

$$x_{2,\text{escalonado}} = \frac{x_2 - \mu_2}{\sigma_2} \rightarrow \text{onde } \sigma_2 \text{ é o desvio padrão de } x_2$$

Supondo $\mu_2 = 2.3$ quartos e $\sigma_2 = 1.4$, teremos $-1.6 \leq x_{2,\text{escalonado}} \leq 1.9$

Exemplos de escalonamento

OPÇÃO 3: Normalização Z-score → Também chamada de **Padronização**

Se $300 \leq x_1 \leq 2000$, podemos escalar x_1 da seguinte maneira:

$$x_{1,\text{escalonado}} = \frac{x_1 - \mu_1}{\sigma_1} \rightarrow \text{onde } \sigma_1 \text{ é o desvio padrão de } x_1$$

Supondo $\mu_1 = 600 \text{ feet}^2$ e $\sigma_1 = 450$, teremos $-0.67 \leq x_{1,\text{escalonado}} \leq 3.1$

Similarmente, se $0 \leq x_2 \leq 5$, podemos escalar x_2 da seguinte maneira:

$$x_{2,\text{escalonado}} = \frac{x_2 - \mu_2}{\sigma_2} \rightarrow \text{onde } \sigma_2 \text{ é o desvio padrão de } x_2$$

Supondo $\mu_2 = 2.3$ quartos e $\sigma_2 = 1.4$, teremos $-1.6 \leq x_{2,\text{escalonado}} \leq 1.9$

Observação:

- Ao dividir uma sequência de números pelo seu desvio padrão, a sequência resultante acaba ficando com desvio unitário.

Intervalos aceitáveis onde não é necessário reescalonar:

- $-1 < x_1 < 1$
- $-3 < x_2 < 3$
- $-0.3 < x_3 < 0.3$
- $0 < x_4 < 3$
- $-2 < x_5 < 0.5$

Intervalos não aceitáveis onde pode ser importante reescalonar:

- $-100 < x_1 < 100$
- $-0.001 < x_2 < 0.001$
- $98.6 < x_3 < 105$

Observação:

Reescalonar quase sempre irá melhorar o desempenho do Método do Gradiente. Raramente irá prejudicar.

Convergência do Método do Gradiente

Perguntas

- Ao rodar o Método do Gradiente, **como saber se ele já convergiu?**
- Qual a relação entre convergência e a taxa de aprendizado α ?

Observação

É importante que consigamos olhar para uma implementação do Método do Gradiente e reconhecer se ela está rodando corretamente ou não.

Convergência do Método do Gradiente

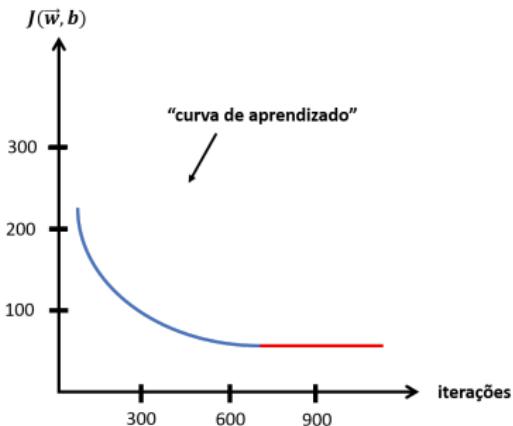
Apenas relembrando que o método do gradiente consiste em repetir até convergir:

$$w_j = w_j - \alpha \frac{d}{dw_j} J(\vec{w}, b)$$

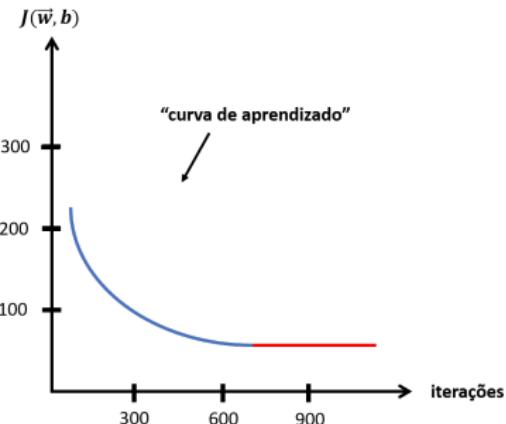
$$b = b - \alpha \frac{d}{db} J(\vec{w}, b)$$

Onde o objetivo dessas atualizações é encontrar os valores de \vec{w} e b que minimizam $J(\vec{w}, b)$.

Portanto, é útil observarmos, ao longo das iterações:



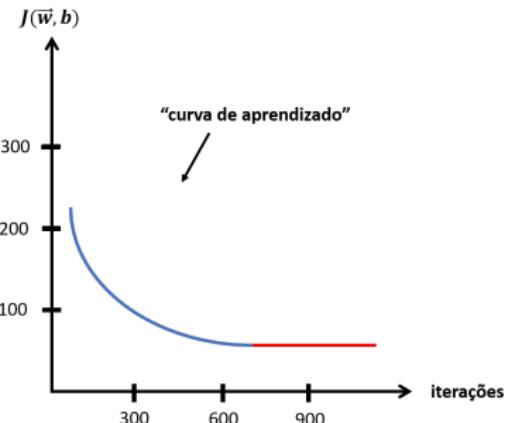
Convergência do Método do Gradiente



Observações:

- Após 300 iterações, o Método do Gradiente ainda está refinando significativamente os valores de w_j e b .
- Após 900 iterações, parece que o método já convergiu.
- O número de iterações que o Método do Gradiente leva para convergir pode variar bastante dependendo da aplicação (30 ou 100000).

Convergência do Método do Gradiente



Observações:

- Após 300 iterações, o Método do Gradiente ainda está refinando significativamente os valores de w_j e b .
- Após 900 iterações, parece que o método já convergiu.
- O número de iterações que o Método do Gradiente leva para convergir pode variar bastante dependendo da aplicação (30 ou 100000).

Observação final:

Após cada iteração, $J(\vec{w}, b)$ deve sempre decrescer. Se isso não ocorrer, então:

- Ou α não foi escolhido apropriadamente (geralmente α muito grande)
- Ou o Método do Gradiente não encontra-se implementado corretamente no código (*bug*)

Convergência do Método do Gradiente

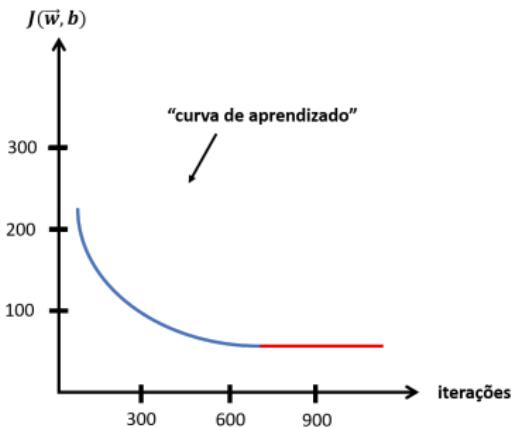
Teste para detecção automática de convergência

Seja ε um valor pequeno, por exemplo, $\varepsilon = 10^{-3}$.

"Se $J(\vec{w}, b)$ decresce menos que ε entre duas iterações consecutivas, então declarar convergência."

Observação

Encontrar um valor adequado para ε pode ser bastante desafiador. Portanto, na dúvida, olhe atentamente o gráfico!



Escolhendo α adequadamente

Observações iniciais

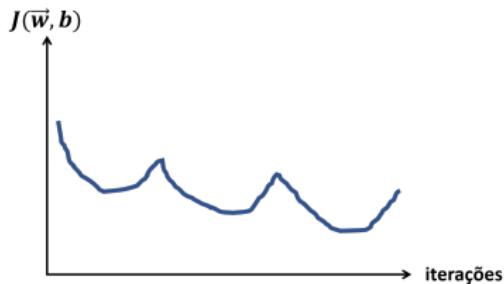
- Se α é muito pequeno, o aprendizado será lento
- Se α é muito grande, o método pode não convergir

Pergunta

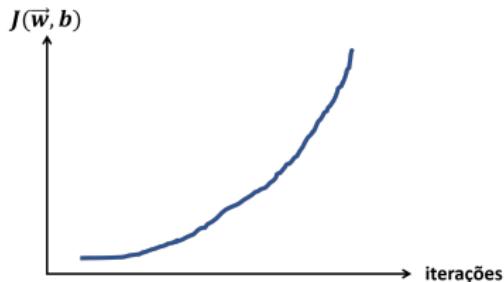
Como escolher um valor adequado para α ?

Escolhendo α adequadamente

Supondo α muito grande, pode acontecer o seguinte:



Supondo um problema de código, por exemplo, $w_j = w_j + \alpha d_j$ pode ocorrer o seguinte:



Dica para debugar o código:

Escolha um valor suficientemente pequeno para α (valor bem pequeno), e verifique se $J(\vec{w}, b)$ está sempre decrescendo iteração após iteração.

Se $J(\vec{w}, b)$ cresce em algum momento, então provavelmente tem-se um bug no código.

Importante:

Usar um valor bem pequeno para α consiste numa boa estratégia para debugar, entretanto fará com que o aprendizado do seu modelo seja lento.

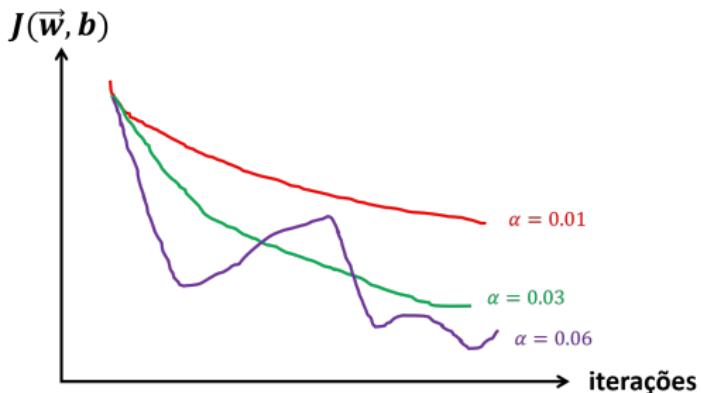
Um método eficiente:

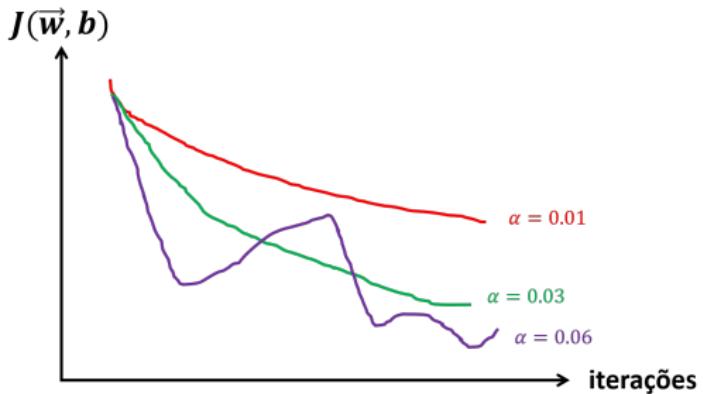
Teste diferentes valores para α

... 0.001 0.01 0.1 1 ...

Para cada valor de α acima, rodar o método do gradiente por um certo número de iterações e:

- verificar para qual escolha de α a função $J(\vec{w}, b)$ decai rapidamente, porém mantendo consistência no seu decaimento.





Observações finais (nível hard)

- A função custo leva em conta todos os parâmetros do modelo simultaneamente. Como todos os parâmetros são atualizados a partir do mesmo α , pode acontecer de um determinado α ser adequado para um parâmetro e não para outro.
- Em alguns raros casos, um determinado valor menor para α pode fazer a função custo decair mais rapidamente a cada iteração em comparação com um α ligeiramente maior. Isso pode acontecer justamente quando esse α não é um valor adequado para certos parâmetros do modelo (o parâmetro encontra-se saltando o seu valor ótimo, por exemplo), ainda que a função custo decaia como um todo a cada iteração.

De olho no código!

De olho no código!

Vamos agora ver como realizar na prática o **escalonamento de características e a escolha do α** .

Acesse o Python Notebook usando o QR code ou o link abaixo:

https://colab.research.google.com/github/xaximpvp2/master/blob/main/codigo_aula7_escalonamento_de_caracteristicas_e_escolha_do_alpha.ipynb



Acesse os dados necessários para rodar o código usando o QR code ou o link abaixo:

https://ufprbr0-my.sharepoint.com/:t/g/personal/ricardo_schumacher_ufpr_br/EVqPY86T5kdAqVvBxfft6EYBM8ppr0_MR5aU6d2l1tT-Wg?e=nL0a7I



OBS: Para adicionar os dados ao ambiente do Colab Notebook, no menu do canto esquerdo da tela do Colab clique em "Arquivos" e depois "Fazer upload para o armazenamento da sessão". Então carregue os arquivos baixados.

Parte 1

Rode todo o código. Certifique-se que você o compreendeu.

Parte 2

- 1** Explique, com suas próprias palavras, como a escolha do α afeta a convergência do Método do Gradiente.
- 2** Explique, com suas próprias palavras, o que é o Escalonamento de Características e qual o seu impacto no Método do Gradiente.

Engenharia de características e Regressão polinomial



- Nas aulas anteriores, mostramos como o Escalonamento das Características impacta o desempenho do Método do Gradiente.

- Nas aulas anteriores, mostramos como o Escalonamento das Características impacta o desempenho do Método do Gradiente.
- Entretanto, vale a pena destacar que a **escolha de quais características devem compor o modelo** também é de fundamental importância.

- Nas aulas anteriores, mostramos como o Escalonamento das Características impacta o desempenho do Método do Gradiente.
- Entretanto, vale a pena destacar que a **escolha de quais características devem compor o modelo** também é de fundamental importância.

Pergunta:

Afinal, **quais características são importantes?**

- Nas aulas anteriores, mostramos como o Escalonamento das Características impacta o desempenho do Método do Gradiente.
- Entretanto, vale a pena destacar que a **escolha de quais características devem compor o modelo** também é de fundamental importância.

Pergunta:

Afinal, **quais características são importantes?**

Resposta:

A área do Aprendizado de Máquina que busca responder a essa pergunta se chama **Engenharia de Características** (*feature engineering*), e vamos falar sobre ela nessa aula.

Exemplo:

Buscando prever o preço de casas, por exemplo, podemos tentar um modelo do tipo

$$f_{\vec{w}, b}(\vec{x}) = w_1 x_1 + w_2 x_2 + b$$

onde

x_1 : Largura do terreno onde a casa está construída

x_2 : Profundidade do terreno onde a casa está construída

Usando nossa intuição, podemos também criar a característica $x_3 = x_1 x_2$ (Área do terreno) e incluí-la no modelo, que passa então a ser:

$$f_{\vec{w}, b}(\vec{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

Engenharia de Características consiste em usar nossa **intuição** ou conhecimento prévio acerca do problema para criar **novas características**, transformando ou combinando outras características já definidas anteriormente.

Importante!

A inclusão de novas características relevantes para o problema pode melhorar significativamente a performance do modelo que está sendo treinado.

Regressão Polinomial

Até o presente momento, focamos em aproximar o comportamento dos nossos dados por meio de **retas**, ou seja, modelos lineares.

Pergunta:

É possível utilizar o Método do Gradiente para ajustar funções polinomiais para os nossos dados?

Até o presente momento, focamos em aproximar o comportamento dos nossos dados por meio de **retas**, ou seja, modelos lineares.

Pergunta:

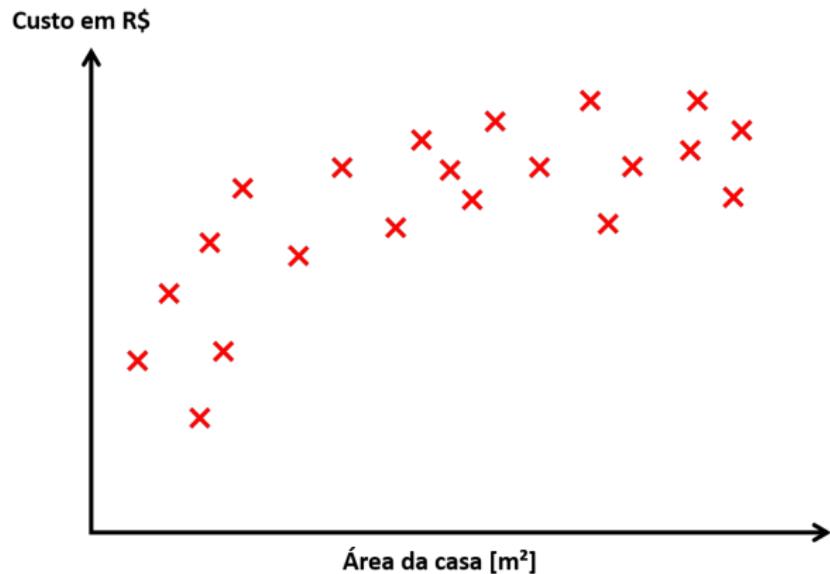
É possível utilizar o Método do Gradiente para ajustar funções polinomiais para os nossos dados?

Resposta:

Sim! Veremos isso agora, onde combinaremos a **regressão linear múltipla** com a **engenharia de características** para criarmos um novo algoritmo, denominado **Regressão Polinomial**.

Exemplo

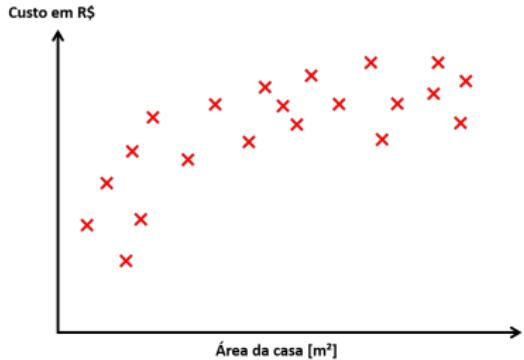
Supondo que você tenha o seguinte conjunto de dados para preços de casas.



Pergunta:

Uma reta é capaz de explicar adequadamente esses dados?

Exemplo



Observação:

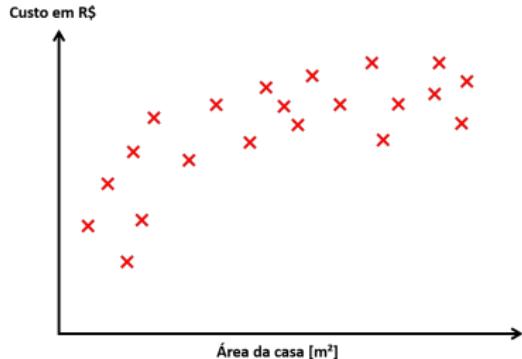
Para esse conjunto de dados, talvez o mais adequado seja tentar uma função quadrática do tipo

$$f(\vec{w}, b)(\vec{x}) = w_1 x + w_2 x^2 + b$$

Pergunta:

Uma parábola parece uma boa ideia?

Exemplo

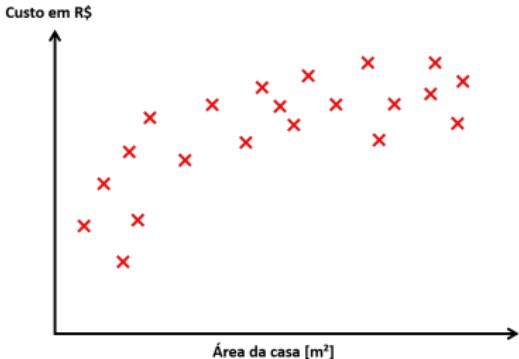


Observação:

Talvez então uma função cúbica do tipo

$$f(\vec{w}, b)(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + b$$

Exemplo



Observação:

Talvez então uma função cúbica do tipo

$$f(\vec{w}, b)(\vec{x}) = w_1 x + w_2 x^2 + w_3 x^3 + b$$

Observação:

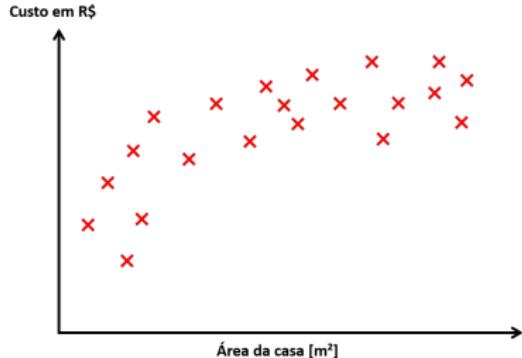
Ao criar características do tipo x^2 , x^3 , ..., o escalonamento de características se torna ainda mais importante:

$$\text{area} = 1 - 10^3$$

$$\text{area}^2 = 1 - 10^6$$

$$\text{area}^3 = 1 - 10^9$$

Exemplo



Uma última hipótese:

Poderíamos também tentar

$$f(\vec{w}, b)(\vec{x}) = w_1 x + w_2 \sqrt{x} + b$$

De olho no código!

De olho no código!

Vamos agora ver um exemplo de código onde é realizada a **engenharia de características** para o contexto de **regressão polinomial**.

Acesse o Python Notebook usando o QR code ou o link abaixo:

https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula8_eng_caracteristicas_e_regressao_polinomial.ipynb



Parte 1

Rode todo o código. Certifique-se que você o compreendeu.

Parte 2

- 1 Crie uma nova função alvo que possui pelo menos um termo polinomial de maior ordem (por exemplo, x^4) e modele-a fazendo as modificações necessárias no código.

Classificação usando Regressão Logística



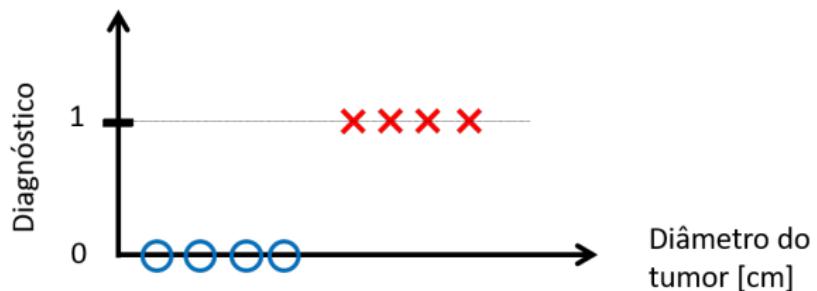
Nas aulas anteriores, aprendemos sobre **Regressão Linear**, onde a variável alvo de saída y podia assumir um **número infinito de valores possíveis**.

Nesta aula e nas próximas, aprenderemos sobre **Classificação**, onde a variável de saída y pode assumir apenas um pequeno conjunto de valores possíveis, denominados “classes”.

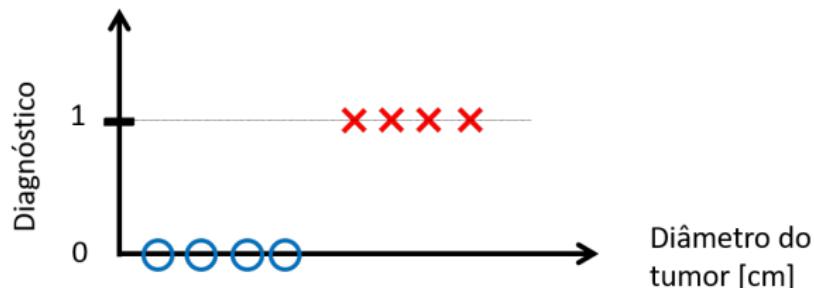
Pergunta	Resposta " y "
Este e-mail é SPAM?	Sim (1) ou Não (0)
Essa transação é fraudulenta?	Sim (1) ou Não (0)
Esse tumor é maligno?	Sim (1) ou Não (0)

- Nos problemas de classificação acima, y pode assumir apenas 1 dentre 2 valores possíveis.
- Problemas desse tipo são denominados **problemas de classificação binária** (apenas 2 possíveis classes/categorias)
- Geralmente, utiliza-se $1 = \text{Sim} = \text{True}$ (classe positiva) e $0 = \text{Não} = \text{False}$ (classe negativa)
- Porém, trata-se apenas de uma convenção. Poderia ser o contrário.

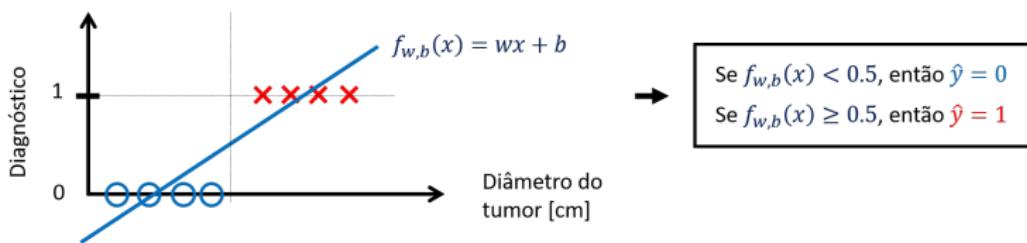
Por que Regressão Linear não é adequada para problemas de classificação?



Por que Regressão Linear não é adequada para problemas de classificação?



Usando Regressão Linear, poderíamos chegar em:



Pergunta:

Parece razoável?

Por que Regressão Linear não é adequada para problemas de classificação?

Adicionando apenas uma amostra de paciente com tumor maligno...

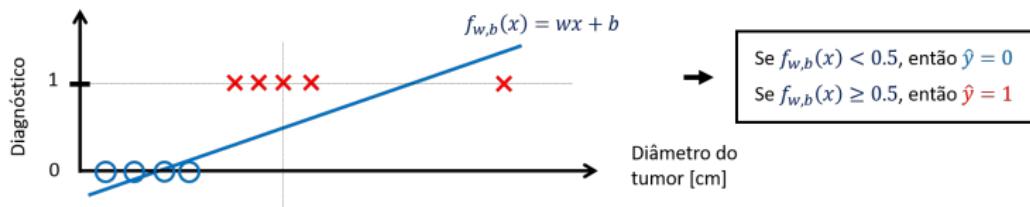


Por que Regressão Linear não é adequada para problemas de classificação?

Adicionando apenas uma amostra de paciente com tumor maligno...



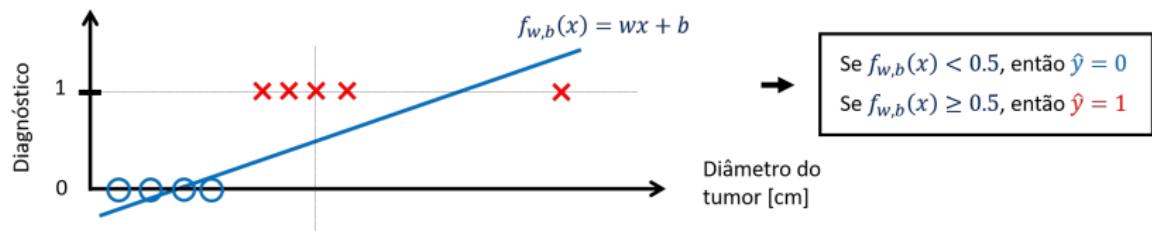
Usando Regressão Linear, poderíamos chegar em:



Observação:

Note que a adição dessa nova amostra não deveria ter mudado a classificação, mas mudou já que o modelo foi treinando via Regressão Linear.

Por que Regressão Linear não é adequada para problemas de classificação?



Observações:

- Note que a Regressão Linear fez com que algumas amostras de tumor maligno fossem classificadas como benignos.
- Isso acontece porque a Regressão Linear não é uma técnica adequada para problemas de classificação
- Para problemas de classificação, temos a **Regressão Logística** que, apesar de ter o termo “regressão” no seu nome, ela não serve para regressão, mas sim para classificação.
- A **Regressão Logística** é um dos algoritmos para classificação mais utilizados, e começaremos por ele.
- A Regressão Logística baseia-se na **função sigmoide**, também chamada de função logística. Essa função sempre fornece valores entre 0 e 1.

Pergunta:

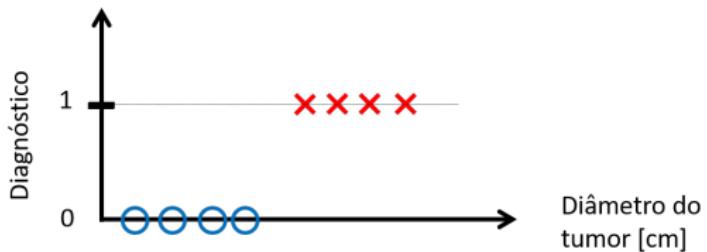
Qual alternativa abaixo representa um problema de classificação?

- A) Estimar o peso de uma baleia com base em seu comprimento
- B) Decidir se um animal é uma baleia ou não.

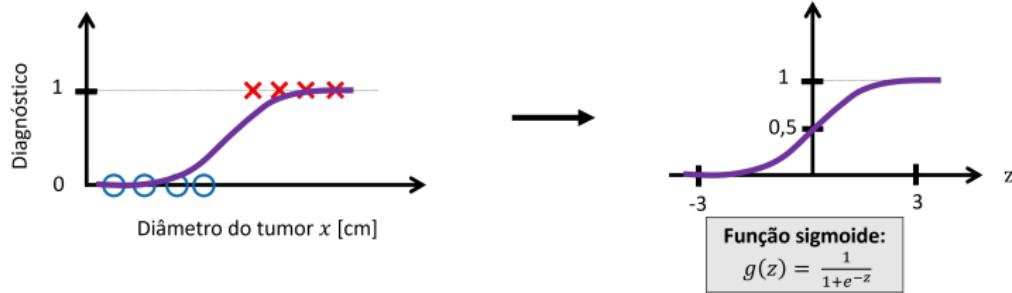
Regressão Logística

- A Regressão Logística é um dos algoritmos de classificação mais simples e utilizados.
- Devido a sua simplicidade, costuma ser um bom método para realização de um teste inicial sobre os seus dados.



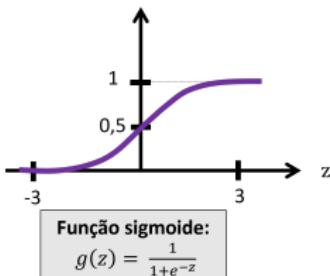


Usando Regressão Logística, podemos chegar em algo do tipo:



- Note que a saída está sempre entre 0 e 1 (por exemplo, 0.7)
- A Regressão Logística baseia-se na **função sigmoide** $g(z)$, também chamada de função logística. Essa função sempre fornece valores entre 0 e 1, ou seja, $0 < g(z) < 1$.

Analizando com detalhes a função sigmoide



Quando z é elevado, por exemplo, $z = 100$, tem-se

$$g(z) \approx \frac{1}{1}$$

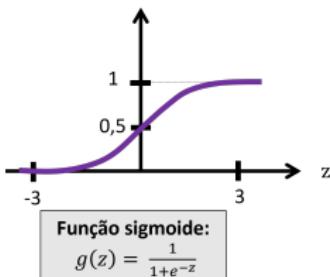
De forma análoga, quando z é bem negativo, por exemplo, $z = -100$, tem-se

$$g(z) \approx \frac{1}{1+\infty} = 0$$

Quando, $z = 0$, tem-se

$$g(z) = \frac{1}{1+1} = 0.5$$

Definindo o modelo de Regressão Logística



Passo 1:

Definimos z como um modelo linear do tipo

$$z = \vec{w} \cdot \vec{x} + b$$

Passo 2:

Passamos este z pela função sigmoide

$$g(z) = \frac{1}{1 + e^{-z}}$$

Resultado

O resultado desse passo-a-passo é o modelo de Regressão Logística

$$f_{\vec{w}, b}(\vec{x}) = g(\vec{w} \cdot \vec{x} + b) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Comparação Regressão Linear com Regressão Logística

Na Regressão Linear, usada em problemas de **Regressão**, tínhamos

$$f_{\vec{w}, b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

Note que $f_{\vec{w}, b}(\vec{x})$ recebia \vec{x} e fornecia valores entre $-\infty$ e $+\infty$

Agora, na Regressão Logística, usada em problemas de **Classificação**, temos

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Note que $f_{\vec{w}, b}(\vec{x})$ agora recebe \vec{x} e fornece valores entre 0 e 1.

Pergunta

- Mas se trata-se de um problema de classificação, onde estão as classes?
- $f_{\vec{w}, b}(\vec{x})$ não deveria fornecer OU 0 OU 1?

Comparação Regressão Linear com Regressão Logística

Na Regressão Linear, usada em problemas de **Regressão**, tínhamos

$$f_{\vec{w}, b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

Note que $f_{\vec{w}, b}(\vec{x})$ recebia \vec{x} e fornecia valores entre $-\infty$ e $+\infty$

Agora, na Regressão Logística, usada em problemas de **Classificação**, temos

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Note que $f_{\vec{w}, b}(\vec{x})$ agora recebe \vec{x} e fornece valores entre 0 e 1.

Pergunta

- Mas se trata-se de um problema de classificação, onde estão as classes?
- $f_{\vec{w}, b}(\vec{x})$ não deveria fornecer OU 0 OU 1?

Resposta

Para resolvemos esse problema, basta interpretarmos $f_{\vec{w}, b}(\vec{x})$ como sendo a **probabilidade** da classe ser 1.

Exemplo

Seja

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

a probabilidade da classe ser 1

Exemplo:

x : tamanho do tumor

y : 1 se o tumor for maligno, ou 0 se o tumor for benigno

$$f_{\vec{w}, b}(\vec{x}) = 0.7 \quad \rightarrow \quad y = 1 \text{ com } 70\% \text{ de chance}$$

$$f_{\vec{w}, b}(\vec{x}) = 0.4 \quad \rightarrow \quad y = 0 \text{ com } 60\% \text{ de chance}$$

Notação formal:

$$f_{\vec{w}, b}(\vec{x}) = P(y = 1 | \vec{x}; \vec{w}, b)$$

Pergunta:

Lembre-se que a função sigmoide é dada por $g(z) = \frac{1}{1+e^{-z}}$. Se z é um número bastante negativo, então:

- A) $g(z)$ é próximo de -1
- B) $g(z)$ é próximo de 0

De olho no código!

Vamos agora ver como implementar a função Sigmoide em Python.

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula9_funcao_sigmoid_e_classificacao.ipynb

Parte 1

Rode todo o código. Certifique-se de que você o compreendeu.

Parte 2

- 1** Explique, com suas próprias palavras, como a função Síntese é utilizada na construção do Método de Regressão Logística.
- 2** Para os dados que estão no código, descubra, por tentativa e erro, quais valores para w e b resultaram num bom modelo de Regressão Logística?

Fronteira de Decisão e Função Custo para Regressão Logística

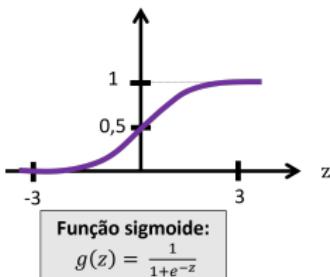


Na aula anterior, revisamos o que são problemas de classificação e fizemos uma introdução ao método de Regressão Logística

Nesta aula continuaremos nossos estudos falando sobre **Fronteira de Decisão**, um conceito importante de ser bem compreendido no âmbito dos algoritmos de classificação.

Também vamos definir a **Função Custo** que utilizaremos para fins de Regressão Logística

Relembrando a Regressão Logística



Passo 1:

Definimos z como um modelo linear do tipo

$$z = \vec{w} \cdot \vec{x} + b$$

Passo 2:

Passamos este z pela função sigmoide

$$g(z) = \frac{1}{1 + e^{-z}}$$

Resultado

O resultado desse passo-a-passo é o modelo de Regressão Logística

$$f_{\vec{w}, b}(\vec{x}) = g(\vec{w} \cdot \vec{x} + b) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}} \quad \text{onde } 0 < f_{\vec{w}, b}(\vec{x}) < 1$$

Gerando as previsões $\hat{y} = 0$ e $\hat{y} = 1$

Como $0 < f_{\vec{w}, b}(\vec{x}) < 1$, podemos considerar que $f_{\vec{w}, b}(\vec{x})$ é a probabilidade de que y seja 1.

Ou seja, $f_{\vec{w}, b}(\vec{x}) = P(y = 1 | \vec{x}; \vec{w}, b)$

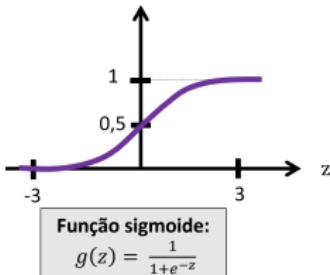
Sendo assim, podemos considerar que

$$\hat{y} = \begin{cases} 1 & \text{se } f_{\vec{w}, b}(\vec{x}) \geq 0.5 \\ 0 & \text{se } f_{\vec{w}, b}(\vec{x}) < 0.5 \end{cases}$$

Pergunta:

Mas quais valores de \vec{w} , b , \vec{x} fazem com que $f_{\vec{w}, b}(\vec{x}) \geq 0.5$? (continua no próximo slide...)

Gerando as previsões $\hat{y} = 0$ e $\hat{y} = 1$



Resposta:

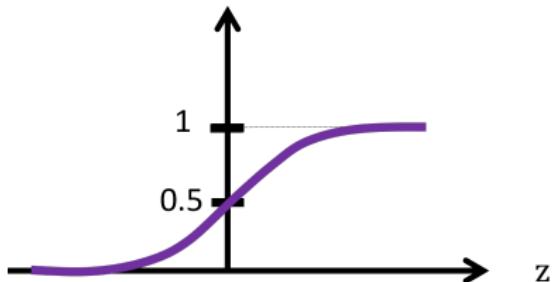
Lembre-se que $f_{\vec{w}, b}(\vec{x}) = g(z)$, sendo $z = \vec{w} \cdot \vec{x} + b$.

Pelo gráfico acima, note que $z \geq 0$ resulta em $g(z) = f_{\vec{w}, b}(\vec{x}) \geq 0.5$

Ou seja, teremos $f_{\vec{w}, b}(\vec{x}) \geq 0.5$ sempre que $z = \vec{w} \cdot \vec{x} + b \geq 0$

Conclusão:

Nosso modelo fará a previsão $\hat{y} = 1$ quando $\vec{w} \cdot \vec{x} + b \geq 0$



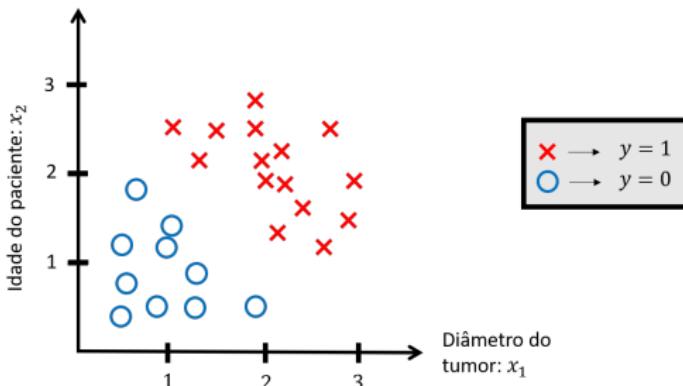
Analogamente, nosso modelo fará a previsão $\hat{y} = 0$ quando $\vec{w} \cdot \vec{x} + b < 0$

Definição:

Com isso, podemos definir a chamada **fronteira de decisão**, que consiste no valor de \vec{x} que faz com que $z = \vec{w} \cdot \vec{x} + b = 0$, ou seja, que faz com que $f_{\vec{w}, b}(\vec{x}) = 0.5$

Definindo a Fronteira de Decisão

Seja o problema de classificação abaixo, que contém 2 variáveis.



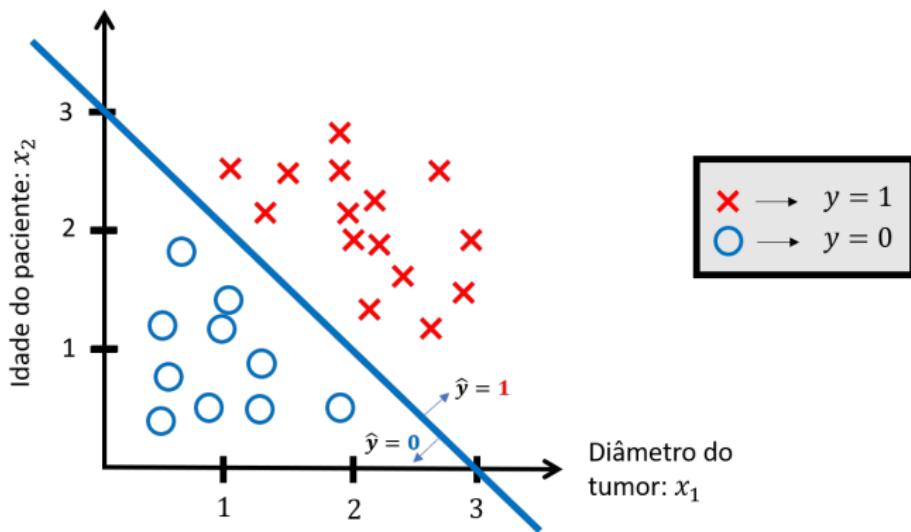
Como são 2 variáveis, o modelo para esse caso é

$$f_{\vec{w}, b}(\vec{x}) = g(w_1 x_1 + w_2 x_2 + b)$$

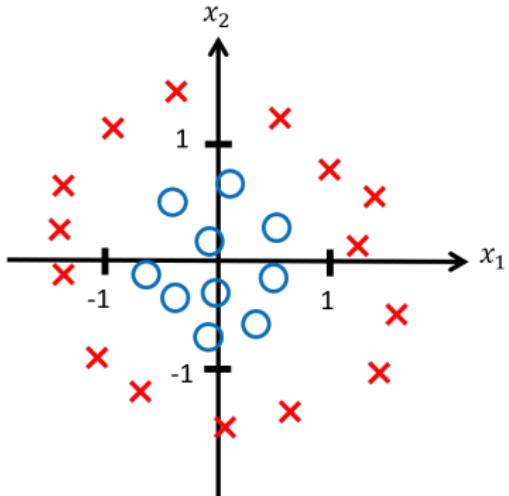
Supondo $w_1 = w_2 = 1$ e $b = -3$, temos a seguinte fronteira de decisão:

$$z = \vec{w} \cdot \vec{x} + b = 0 \quad \rightarrow \quad x_1 + x_2 - 3 = 0 \quad \rightarrow \quad x_1 + x_2 = 3$$

Ilustrando a fronteira de decisão $x_1 + x_2 = 3$ na figura, temos



Podemos ter também Fronteiras de Decisão mais complexas (não lineares):



Usando **Engenharia de Características**, podemos estabelecer o seguinte modelo para esse caso:

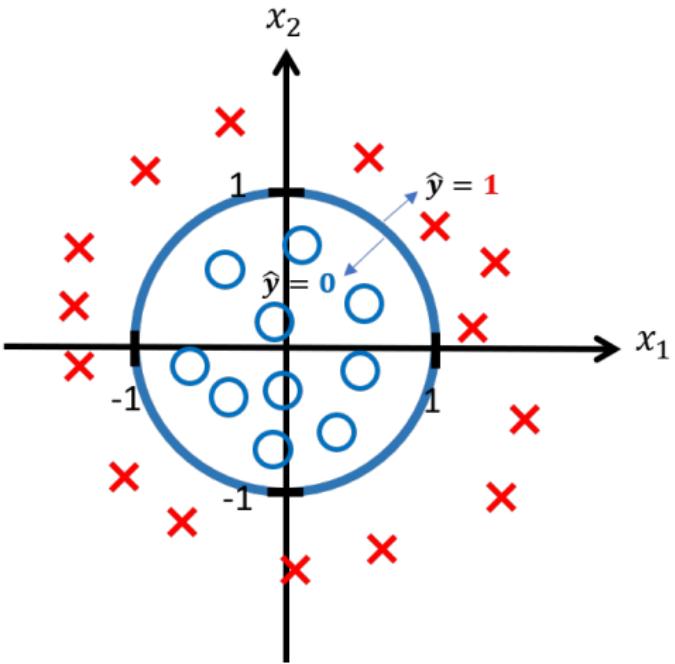
$$f_{\vec{w}, b}(\vec{x}) = g(w_1 x_1^2 + w_2 x_2^2 + b)$$

Supondo $w_1 = w_2 = 1$ e $b = -1$, temos a seguinte fronteira de decisão:

$$z = \vec{w} \cdot \vec{x} + b = 0 \quad \rightarrow \quad x_1^2 + x_2^2 - 1 = 0 \quad \rightarrow \quad x_1^2 + x_2^2 = 1$$

Podemos ter também Fronteiras de Decisão mais complexas (não lineares):

Ilustrando a fronteira de decisão $x_1^2 + x_2^2 = 1$ na figura, temos



Observação

- Usando polinômios de maior ordem, podemos ter fronteiras de decisão com formas ainda mais complexas.
- Ou seja, assim como na Regressão Linear nós não estávamos limitados a estimar retas para os dados, aqui na Regressão Logística também não estamos limitados a estimar Fronteiras de Decisão lineares.
- Apesar disso, continua valendo a relação linear

$$z = \vec{w} \cdot \vec{x} + b$$

Pergunta:

Suponha que você está criando um algoritmo para classificação de tumor. Seu algoritmo será usado para sinalizar potenciais tumores que deverão ser investigados em mais detalhes por um especialista. Qual valor de limiar você utilizaria?

- A) Um valor elevado, de 0.9
- B) Um valor baixo, de 0.2

Função Custo para Regressão Logística

Função Custo para Regressão Logística

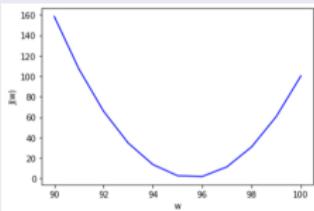
Na Regressão linear, tínhamos a seguinte função custo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2$$

onde

$$f_{\vec{w}, b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

Vimos que, com essas definições, a função quadrática J é **convexa**, ou seja, ela não possui outros mínimos locais além do próprio mínimo global:

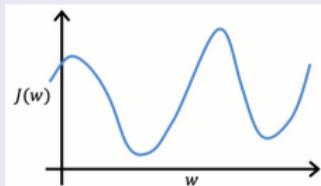


Agora, na Regressão Logística, temos:

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Pergunta: Agora para esse novo $f_{\vec{w}, b}(\vec{x})$, é uma boa ideia usarmos a mesma definição para $J(\vec{w}, b)$?

Resposta: Não, afinal, é possível provar que, se fizermos isso, $J(\vec{w}, b)$ **não será convexa**, e terá diversos mínimos locais, aos quais podemos ficar presos.



Conclusão:

Precisaremos modificar um pouco nossa função custo $J(\vec{w}, b)$ para que ela se torne convexa agora com

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

A função custo quadrática da **Regressão Linear** pode ser reescrita conforme segue:

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$$

onde $L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$ é chamada de "**função de perda**" e é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \frac{1}{2} (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2$$

Função Custo para Regressão Logística

A função custo quadrática da **Regressão Linear** pode ser reescrita conforme segue:

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$$

onde $L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$ é chamada de "**função de perda**" e é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \frac{1}{2} (f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)})^2$$

Na **Regressão Logística**, também usamos a função custo:

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$$

Porém, na **Regressão Logística**, a "**função de perda**" é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 0 \end{cases}$$

OBS: Tal função é também chamada de "**função de entropia cruzada binária**".

Observação adicional: Estamos usando a notação $\log = \ln$ (logaritmo neperiano)

Na Regressão Logística, a **função de perda** é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 0 \end{cases}$$

Observações:

- a função de perda $L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$ opera com base em uma única amostra i
- Para obter a função custo, é necessário somar a perda para todas as amostras e depois dividir por m

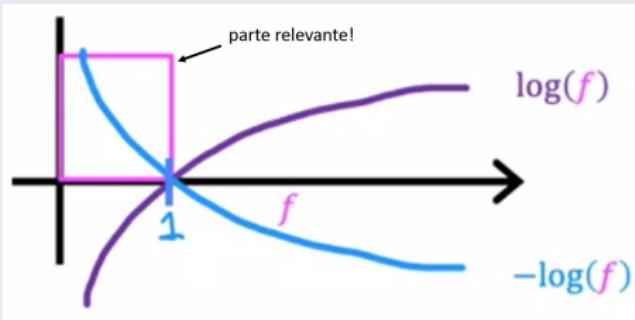
Analizando melhor a função perda

Na Regressão Logística, a **função de perda** é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 0 \end{cases}$$

Analisando o caso $y^{(i)} = 1$

Como $f_{\vec{w}, b}(\vec{x}^{(i)})$ estará sempre entre 0 e 1, então a única parte relevante da função $-\log(f_{\vec{w}, b}(\vec{x}^{(i)}))$ para nós é:



- Ou seja, a **perda** $\rightarrow 0$ conforme $f_{\vec{w}, b}(\vec{x}^{(i)}) \rightarrow 1 = y^{(i)}$. Ou seja, à medida com que a previsão torna-se correta! Por outro lado, **perda** $\rightarrow \infty$ conforme $f_{\vec{w}, b}(\vec{x}^{(i)}) \rightarrow 0$.

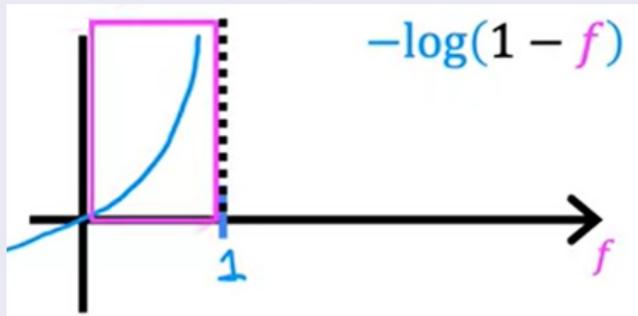
Analizando melhor a função perda

Na Regressão Logística, a **função de perda** é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 0 \end{cases}$$

Analizando agora o caso $y^{(i)} = 0$

Como nossa $f_{\vec{w}, b}(\vec{x}^{(i)})$ estará sempre entre 0 e 1, então a única parte relevante da função $-\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)}))$ para nós é:



- Ou seja, a **perda** $\rightarrow 0$ conforme $f_{\vec{w}, b}(\vec{x}^{(i)}) \rightarrow 0 = y^{(i)}$. Ou seja, à medida com que a previsão torna-se correta! Por outro lado, **perda** $\rightarrow \infty$ conforme $f_{\vec{w}, b}(\vec{x}^{(i)}) \rightarrow 1$.

Analisando melhor a função perda

Na Regressão Logística, a função de perda é dada por

$$L(f_{\vec{w}}, b)(\vec{x}^{(i)}, y^{(i)}) = \begin{cases} -\log(f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 0 \end{cases}$$

- Definindo a função perda dessa forma, o modelo sofre uma penalidade zero (nula) quando sua previsão $f_{\vec{w}, b}(\vec{x}^{(i)})$ para uma determinada amostra $y^{(i)}$ está correta.
- A penalidade sofrida pelo modelo aumenta conforme sua previsão $f_{\vec{w}, b}(\vec{x}^{(i)})$ se afasta do valor alvo $y^{(i)}$
- Ao errar completamente a previsão, por exemplo, $y^{(i)} = 1$ e $f_{\vec{w}, b}(\vec{x}^{(i)}) = 0$, o modelo é drasticamente penalizado (perda $\rightarrow \infty$)
- A função custo é a soma das perdas para todas as m amostras, divididos por m :

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$$

- O objetivo é encontrar os valores de \vec{w}, b que minimizam essa perda média, $J(\vec{w}, b)$.
- É possível provar que essa função $J(\vec{w}, b)$ será convexa para

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

- Ou seja, ela possui apenas um mínimo, sendo este, portanto, seu mínimo global.

De olho no código!

De olho no código!

Vamos agora explorar o conceito de Fronteira de Decisão e Função Custo para Regressão Logística.

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula10_fronteira_de_decisao.ipynb

Parte 1

Rode todo o código. Certifique-se de que você o compreendeu.

Parte 2

- 1 Explique, com suas próprias palavras, o conceito de Fronteira de Decisão no contexto da Regressão Logística.
- 2 Considerando $w_0 = w_1 = 1$ e $b = -3$, calcule o valor do modelo $f_{\vec{w}, b}(\vec{x}^{(i)})$ para cada amostra de dados $\vec{x}^{(i)}$ presente no código. O que esses valores representam? Os resultados estão coerentes com aquilo que é observado graficamente no código?
- 3 Calcule o valor da função perda para cada amostra i .
- 4 Calcule o custo $J(\vec{w}, b)$.

Regressão Logística: Função Custo Simplificada e Método do Gradiente



Na aula anterior, definimos a **função perda** e a **função custo** para a Regressão Logística.

Nesta aula, iremos simplificar a definição da função custo e aplicar o Método do Gradiente para encontrar seu mínimo global.

Na aula passada, vimos que a **Função de perda** para Regressão Logística é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 0 \end{cases}$$

OBS: $y^{(i)}$ é sempre 1 ou 0 (tumor maligno ou não)

Na aula passada, vimos que a **Função de perda** para Regressão Logística é dada por

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -\log(f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 1 \\ -\log(1 - f_{\vec{w}, b}(\vec{x}^{(i)})) & , \text{ se } y^{(i)} = 0 \end{cases}$$

OBS: $y^{(i)}$ é sempre 1 ou 0 (tumor maligno ou não)

Note que é possível simplificar, escrevendo da seguinte forma equivalente:

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = -y^{(i)} \log(f_{\vec{w}, b}(\vec{x}^{(i)})) - (1 - y^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}^{(i)}))$$

Pergunta:

Como a função de perda abaixo é simplificada para o caso em que a amostra i correspondente possui valor alvo $y^{(i)} = 1$?

$$L\left(f_{\vec{w}, b}\left(\vec{x}^{(i)}\right), y^{(i)}\right) = -y^{(i)} \log \left(f_{\vec{w}, b}\left(\vec{x}^{(i)}\right)\right) - (1 - y^{(i)}) \log \left(1 - f_{\vec{w}, b}\left(\vec{x}^{(i)}\right)\right)$$

- A) $-\log \left(f_{\vec{w}, b}\left(\vec{x}^{(i)}\right)\right)$
- B) $-\log \left(f_{\vec{w}, b}\left(1 - \vec{x}^{(i)}\right)\right)$

Função custo simplificada

A função custo que queremos minimizar na **Regressão Logística** é a média das perdas:

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$$

onde

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = -y^{(i)} \log(f_{\vec{w}, b}(\vec{x}^{(i)})) - (1 - y^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}^{(i)}))$$

Sendo assim, note que podemos reescrever

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(f_{\vec{w}, b}(\vec{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}^{(i)}))]$$

Função custo simplificada

A função custo que queremos minimizar na **Régressão Logística** é a média das perdas:

$$J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)})$$

onde

$$L(f_{\vec{w}, b}(\vec{x}^{(i)}), y^{(i)}) = -y^{(i)} \log(f_{\vec{w}, b}(\vec{x}^{(i)})) - (1 - y^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}^{(i)}))$$

Sendo assim, note que podemos reescrever

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(f_{\vec{w}, b}(\vec{x}^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\vec{w}, b}(\vec{x}^{(i)}))]$$

Observação:

- O Método do Gradiente agora pode ser aplicado para encontrar quais são os parâmetros \vec{w} , b que minimizam $J(\vec{w}, b)$
- Lembrando que $J(\vec{w}, b)$ será convexa (único mínimo), ainda que o modelo seja:

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

onde $f_{\vec{w}, b}(\vec{x})$ é a probabilidade de y ser 1.

Precisamos encontrar os valores de \vec{w} , b que minimizam

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) + (1 - y^{(i)}) \log \left(1 - f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) \right]$$

Precisamos encontrar os valores de \vec{w} , b que minimizam

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) + (1 - y^{(i)}) \log \left(1 - f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) \right]$$

Sabemos que o Método do Gradiente consiste em repetir:

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b) \quad j = 1, \dots, n$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

Precisamos encontrar os valores de \vec{w} , b que minimizam

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) + (1 - y^{(i)}) \log \left(1 - f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) \right]$$

Sabemos que o Método do Gradiente consiste em repetir:

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b) \quad j = 1, \dots, n$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

É possível provar que, para a função $J(\vec{w}, b)$ acima, tem-se:

$$\frac{\partial}{\partial w_j} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) - y^{(i)} \right) x_j^{(i)}$$

$$\frac{\partial}{\partial b} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) - y^{(i)} \right)$$

$L(f_{\tilde{w},b}(\vec{x}), y) = -y \log(f_{\tilde{w},b}(\vec{x})) - (1-y) \log(1-f_{\tilde{w},b}(\vec{x}))$

 $\frac{\partial L}{\partial w_j} = -y \left[\frac{\partial}{\partial w_j} \log(f_{\tilde{w},b}(\vec{x})) \right] - (1-y) \left[\frac{\partial}{\partial w_j} \log(1-f_{\tilde{w},b}(\vec{x})) \right]$
 $f_{\tilde{w},b}(\vec{x}) = \frac{1}{1+e^{-(\tilde{w}\vec{x}+b)}} = (1+e^{-(\tilde{w}\vec{x}+b)})^{-1}$
 $\frac{\partial}{\partial w_j} \log(f_{\tilde{w},b}(\vec{x})) = \frac{1}{f_{\tilde{w},b}(\vec{x})} \cdot \frac{\partial}{\partial w_j} f_{\tilde{w},b}(\vec{x}) = e^{-(\tilde{w}\vec{x}+b)} \cdot (-x_j)$
 $\frac{\partial}{\partial w_j} f_{\tilde{w},b}(\vec{x}) = \frac{\partial}{\partial w_j} (1+e^{-(\tilde{w}\vec{x}+b)})^{-1} = -(1+e^{-(\tilde{w}\vec{x}+b)})^{-2} \cdot \frac{\partial}{\partial w_j} e^{-(\tilde{w}\vec{x}+b)}$
 $= (1+e^{-(\tilde{w}\vec{x}+b)})^{-2} \cdot \frac{-e^{-(\tilde{w}\vec{x}+b)}}{e^{-(\tilde{w}\vec{x}+b)}} \cdot x_j$
 $\frac{\partial}{\partial w_j} \log(f_{\tilde{w},b}(\vec{x})) = \frac{(1+e^{-(\tilde{w}\vec{x}+b)}) \cdot (1+e^{-(\tilde{w}\vec{x}+b)})^{-2} \cdot e^{-(\tilde{w}\vec{x}+b)} \cdot (-x_j)}{(1+e^{-(\tilde{w}\vec{x}+b)}) + (1+e^{-(\tilde{w}\vec{x}+b)})^{-1} \cdot x_j}$
 $\frac{\partial}{\partial w_j} \log(1-f_{\tilde{w},b}(\vec{x})) = \frac{1}{1-f_{\tilde{w},b}(\vec{x})} \cdot \left(\frac{\partial}{\partial w_j} f_{\tilde{w},b}(\vec{x}) \right)$
 $= \frac{1}{1-\frac{1}{1+e^{-(\tilde{w}\vec{x}+b)}}} \cdot \frac{(1+e^{-(\tilde{w}\vec{x}+b)})^2 \cdot e^{-(\tilde{w}\vec{x}+b)} \cdot x_j}{(1+e^{-(\tilde{w}\vec{x}+b)}) - \frac{1}{1+e^{-(\tilde{w}\vec{x}+b)}}}$
 $\frac{\partial L}{\partial w_j} = -y \left[\frac{e^{-(\tilde{w}\vec{x}+b)}}{1+e^{-(\tilde{w}\vec{x}+b)}} \cdot x_j \right] + (1-y) \left[\frac{x_j}{1+e^{-(\tilde{w}\vec{x}+b)}} \right]$
 $= x_j \left[-y \frac{e^{-(\tilde{w}\vec{x}+b)}}{1+e^{-(\tilde{w}\vec{x}+b)}} - y \frac{1}{1+e^{-(\tilde{w}\vec{x}+b)}} + \frac{1+e^{-(\tilde{w}\vec{x}+b)}}{1+e^{-(\tilde{w}\vec{x}+b)}} \right] = x_j \left(-y + f_{\tilde{w},b}(\vec{x}) \right)$

① $-y \frac{1+e^{-(\tilde{w}\vec{x}+b)}}{1+e^{-(\tilde{w}\vec{x}+b)}} = -y$

Método do Gradiente aplicado à Regressão Logística

Método do Gradiente para Regressão Logística consiste em repetir:

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right) \right]$$

Parece idêntico à Regressão Linear. Porém, devemos lembrar que:

Regressão linear:

$$f_{\vec{w}, b} (\vec{x}^{(i)}) = \vec{w} \cdot \vec{x} + b$$

Regressão Logística:

$$f_{\vec{w}, b} (\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Quando estudamos Regressão Linear, é recomendado:

- Monitorar o Método do Gradiente (curva de aprendizado)
- Realizar implementação Vetorizada
- Escalonamento de Características

Todos estes conceitos continuam valendo agora para o contexto de Regressão Logística

De olho no código!

De olho no código!

Vamos agora implementar o algoritmo de Regressão Logística na prática.

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula_11_regressao_logistica.ipynb

Parte 1

Rode todo o código. Certifique-se de que você o compreendeu.

Parte 2

- 1** Explique, com suas próprias palavras, quais são os passos necessários para implementar a Regressão Logística na prática.
- 2** Faça modificações no conjunto de dados que está no código, e verifique como essas modificações alteram a fronteira de decisão do modelo estimado.

Regularização para redução da sobreestimação



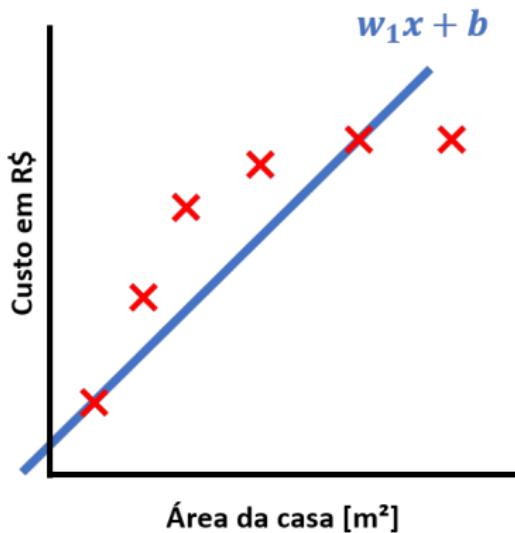
Na aula anterior, implementamos o algoritmo de **Regressão Logística**.

Nesta aula, vamos aprender sobre **sobreestimação**, também conhecida pelo termo *overfitting*, sendo esse um problema comum que nosso modelo pode apresentar em algumas situações.

Pergunta:

Mas afinal, o que é *overfitting*?

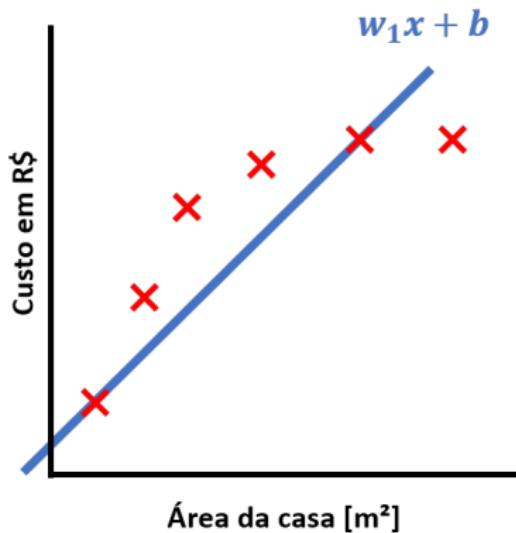
Um exemplo vindo da Regressão



Perguntas

- 1 O modelo acima se ajusta bem aos dados?

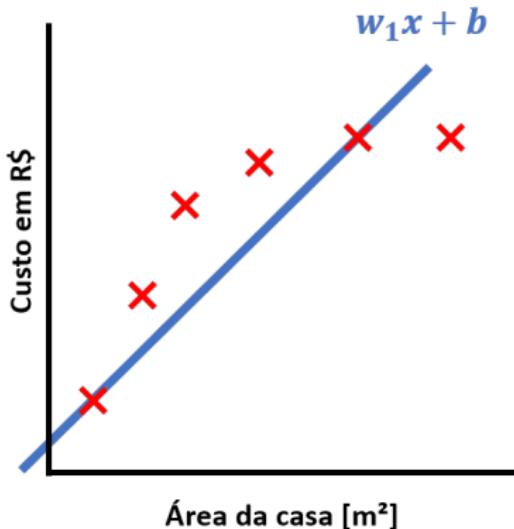
Um exemplo vindo da Regressão



Perguntas

- 1 O modelo acima se ajusta bem aos dados?
- 2 O modelo subestima ou sobreestima os dados?

Um exemplo vindo da Regressão



Perguntas

- 1 O modelo acima se ajusta bem aos dados?
- 2 O modelo subestima ou sobreestima os dados?

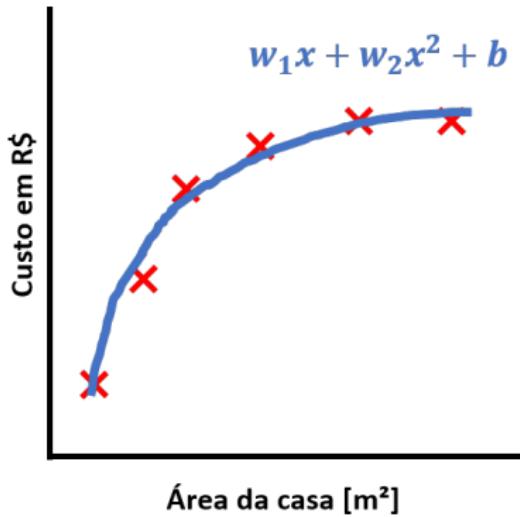
Termos:

underfit = high bias



O modelo não é capaz de explicar suficientemente o comportamento presente nos dados.

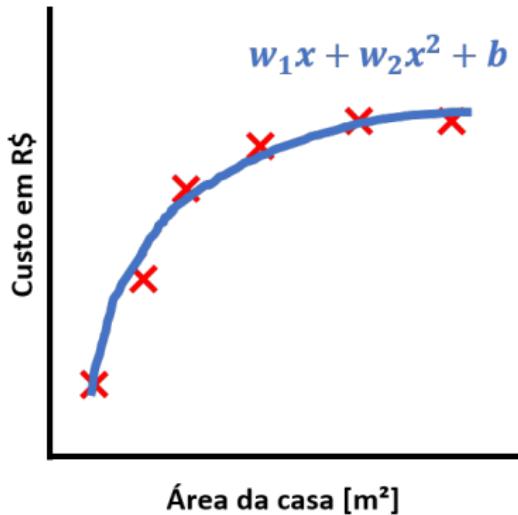
Um exemplo vindo da Regressão



Perguntas

- 1 Esse segundo modelo se ajusta bem aos dados?

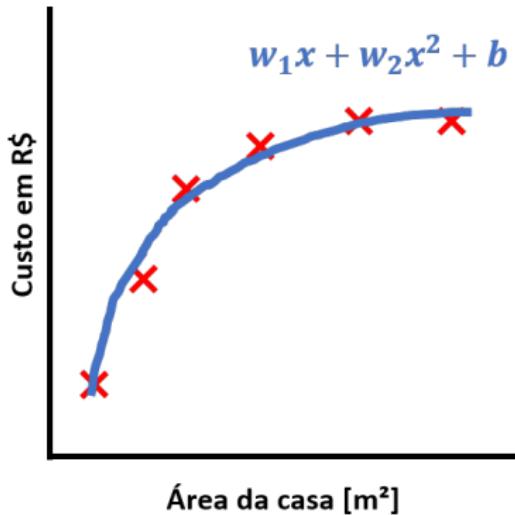
Um exemplo vindo da Regressão



Perguntas

- 1 Esse segundo modelo se ajusta bem aos dados?
- 2 O modelo subestima ou sobreestima os dados?

Um exemplo vindo da Regressão



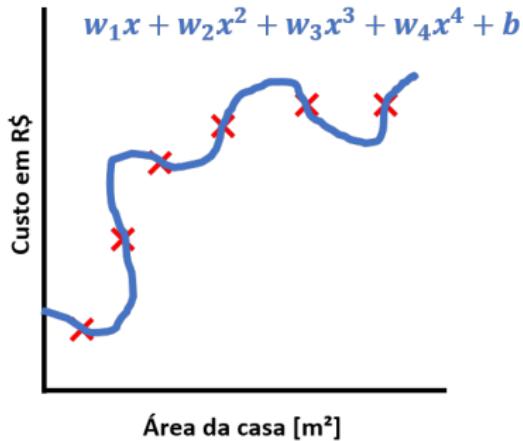
Perguntas

- 1 Esse segundo modelo se ajusta bem aos dados?
- 2 O modelo subestima ou sobreestima os dados?

Termos:

Generalização → é a capacidade que um modelo tem (ou não) de performar bem para dados não usados durante seu treinamento.

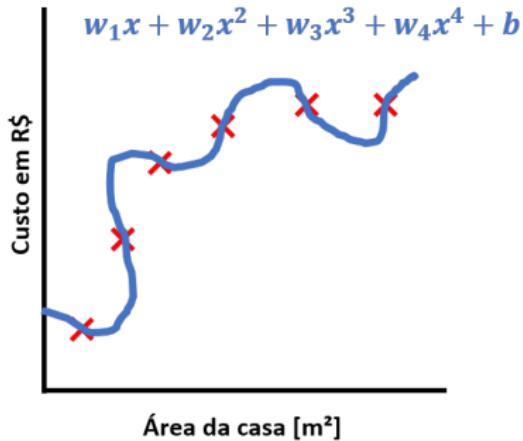
Um exemplo vindo da Regressão



Perguntas

- 1 Esse terceiro modelo se ajusta perfeitamente aos dados de treinamento?

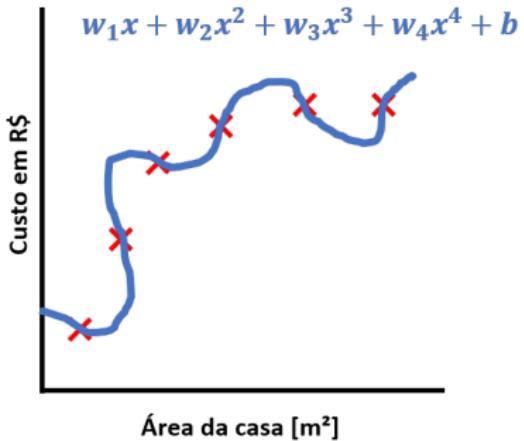
Um exemplo vindo da Regressão



Perguntas

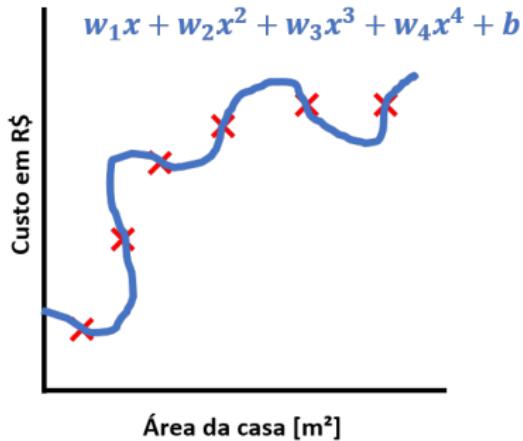
- 1 Esse terceiro modelo se ajusta perfeitamente aos dados de treinamento?
- 2 Qual seria o valor da função custo $J(\vec{w}, b)$ para esse caso?

Um exemplo vindo da Regressão



Perguntas

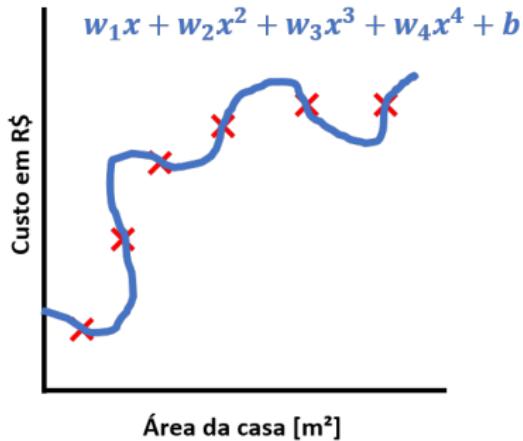
- 1 Esse terceiro modelo se ajusta perfeitamente aos dados de treinamento?
- 2 Qual seria o valor da função custo $J(\vec{w}, b)$ para esse caso?
- 3 Qual é o problema com esse modelo então?



Perguntas

- 1 Esse terceiro modelo se ajusta perfeitamente aos dados de treinamento?
- 2 Qual seria o valor da função custo $J(\vec{w}, b)$ para esse caso?
- 3 Qual é o problema com esse modelo então?
- 4 O modelo subestima ou sobreestima os dados?

Um exemplo vindo da Regressão



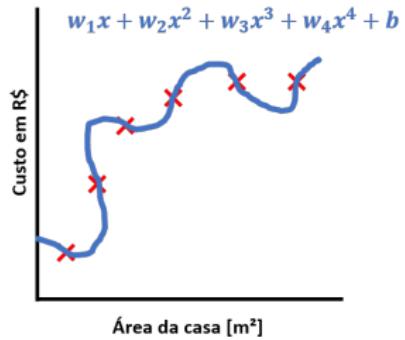
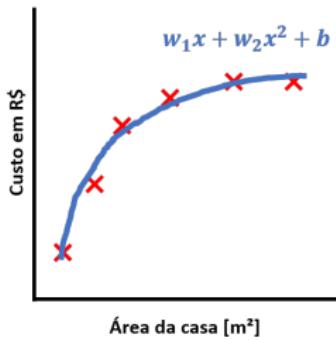
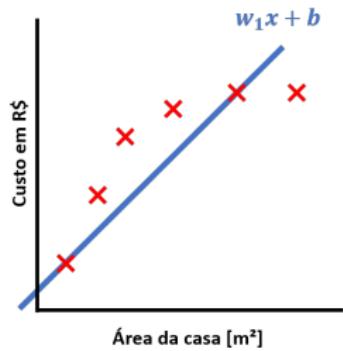
Perguntas

- 1 Esse terceiro modelo se ajusta perfeitamente aos dados de treinamento?
- 2 Qual seria o valor da função custo $J(\vec{w}, b)$ para esse caso?
- 3 Qual é o problema com esse modelo então?
- 4 O modelo subestima ou sobreestima os dados?

Termos:

overfit = high variance → O modelo se ajustou mais do que deveria aos dados que lhe foram mostrados.

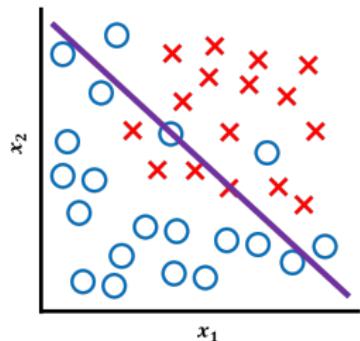
Um exemplo vindo da Regressão



Pergunta

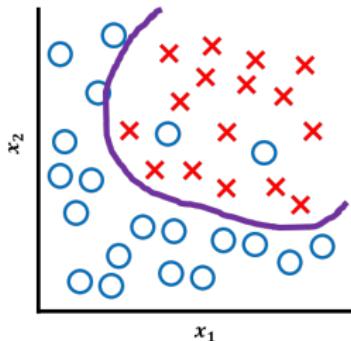
Qual dos três modelos acima você escolheria?

- Modelos muito simples, com poucos parâmetros, podem não ser suficientes para explicar o comportamento presente nos dados
- Por outro lado, modelos muito complexos, com um excesso de parâmetros, podem explicar perfeitamente bem os dados de treinamento, mas não generalizar bem para novos dados

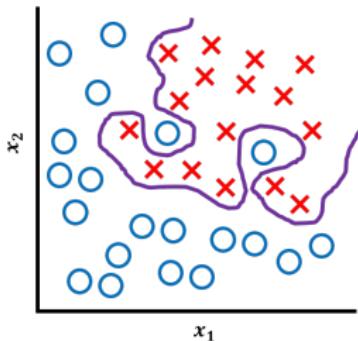


$$z = w_1x_1 + w_2x_2 + b$$

$$f_{\vec{w}, b}(\vec{x}) = g(z)$$



$$z = w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$$

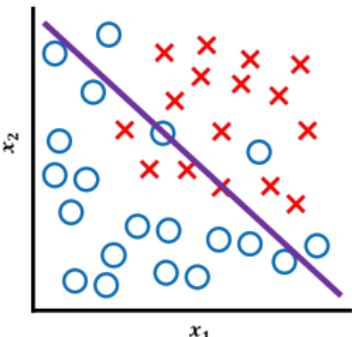


$$z = w_1x_1 + w_2x_2 + w_3x_1^2x_2 + w_4x_1^2x_2^2 + w_5x_1^2x_2^3 + w_6x_1^3x_2 + \dots + b$$

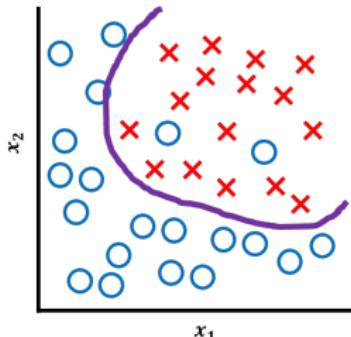
Perguntas

- 1 Qual modelo subestima os dados?
- 2 Qual modelo parece super ok?
- 3 Qual modelo sobreestima os dados?

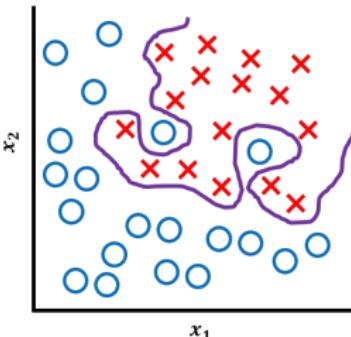
As mesmas conclusões se aplicam para problemas de **classificação**



$$z = w_1x_1 + w_2x_2 + b$$
$$f_{\vec{w},b}(\vec{x}) = g(z)$$



$$z = w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$$



$$z = w_1x_1 + w_2x_2 + w_3x_1^2x_2 + w_4x_1^2x_2^2 + w_5x_1^2x_2^3 + w_6x_1^3x_2 + \dots + b$$

Pergunta

Seja x_1 : diâmetro do tumor e x_2 : idade do paciente.

Qual dos três modelos acima você escolheria para estimar a probabilidade de um novo paciente estar ou não com um tumor maligno?

Pergunta:

O nosso objetivo é criar um modelo capaz de prever valores de saída corretamente para **novas amostras**, ou seja, que **generalize bem**.

Quando um modelo se ajuste bem aos dados de treinamento, mas não funciona bem para novas amostras que não estavam presentes no seu treinamento, isso é exemplo de:

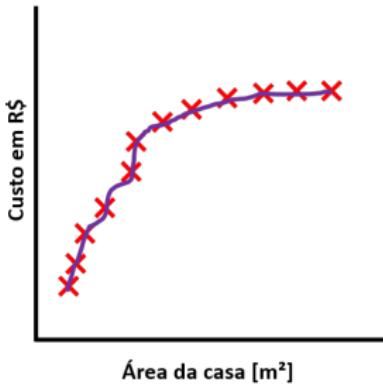
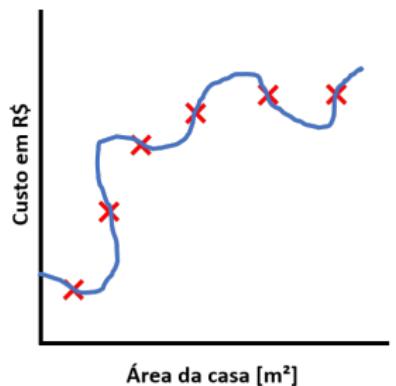
- A) Sobreestimação
- B) Subestimação
- C) Um modelo que generaliza bem

Como resolver o problema de overfitting?

Como resolver o problema de overfitting?

Opção 1

Colete e utilize mais dados durante o treinamento:



Observação

Infelizmente, coletar mais dados nem sempre é uma opção.

Como resolver o problema de overfitting?

Opção 2

Selecione as características mais relevantes:



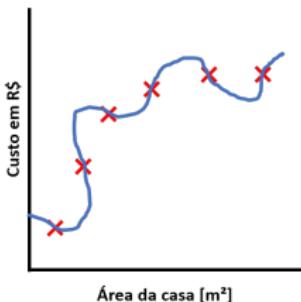
Observações

- Muitas características + poucos dados podem levar à sobreestimação.
- Use a intuição para selecionar: distância até o mercado mais próximo é de fato importante?
- **Desvantagem:** Características relevantes podem ser ignoradas (informação relevante perdida).

Como resolver o problema de overfitting?

Opção 3

Regularização



$$f(x) = 28x - 385x^2 + 39x^3 - 174x^4 + 100$$

- Em muitos casos, o overfitting ocorre pois alguns parâmetros do modelo assumem valores muito elevados (exemplo: $w_2 = -385$ e $w_4 = -174$)
- Regularização permite que os parâmetros existam, mas gera uma penalização elevada caso eles sejam excessivamente elevados.
- Olhando o caso específico acima, é esperado que w_2 tenha de fato um valor ligeiramente elevado, pois ele será responsável pelo “comportamento de parábola” passível de ser observado nos dados. Entretanto, não é esperado que w_3 e w_4 sejam elevados.
- Em problemas mais complexos, isso não é visual, e teremos que lidar com isso de uma forma mais generalizada e sistematizada.
- Nesse sentido, em geral, regularizamos apenas os parâmetros w_j do modelo.
- Regularizar também o parâmetro b geralmente não gera muito impacto.

Como resolver o problema de overfitting?

Opção 1

Coletar mais dados

Opção 2

Selecionar as características

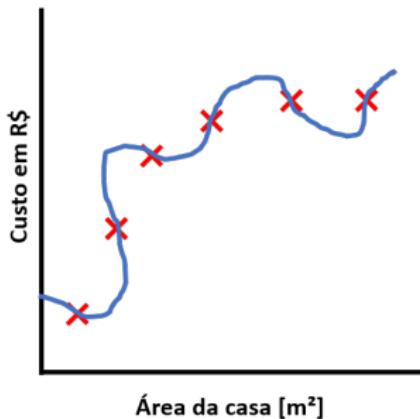
Opção 3

Regularização → Estudaremos agora com mais detalhes como implementar!

Veremos agora como implementar a Regularização na prática

Implementando a Regularização

No exemplo abaixo, se escolhermos valores excessivamente grandes para w_3 e w_4 podemos ter overfitting.



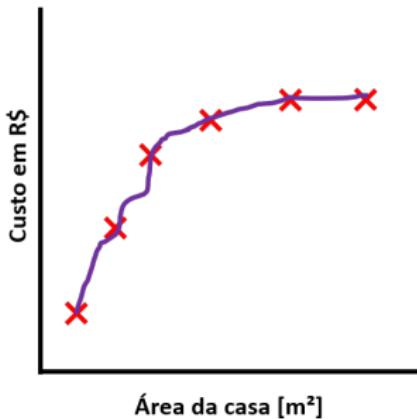
$$f(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

Pergunta:

O que acontece se estimarmos os parâmetros \vec{w}, b por meio da função custo modificada:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + 1000w_3^2 + 1000w_4^2 ?$$

OBS: Note que estamos penalizando valores elevados para w_3 e w_4 multiplicando ambos por um valor escalar elevado e adicionando esses termos à função custo $J(\vec{w}, b)$.



$$f(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

Resposta:

- Os parâmetros w_3 e w_4 serão garantidamente pequenos, 0.001 e 0.002 por exemplo.
- Com isso, a chance de overfitting é drasticamente reduzida.
- Por outro lado, w_3 e w_4 ainda permanecem presentes no modelo, contribuindo para que o modelo explique bem os dados.
- Na prática, penalizamos todos os parâmetros w_j do modelo, para $j = 1, \dots, n$
- Geralmente isso leva a modelos com resposta mais suave e que não sobreestimam os dados.

Implementando a Regularização (caso geral)

Área da casa [m ²] (x_1)	Número de quartos (x_2)	Idade [anos] (x_3)	...	Distância até mercado (x_{100})	Custo (y)

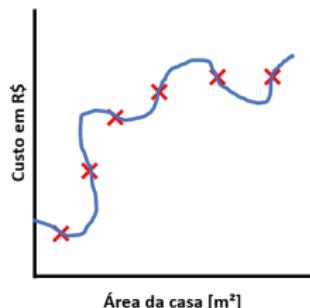
No caso geral, como não sabemos quais características são mais importantes, penalizamos todos os parâmetros w_j , usando a função custo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b} (\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

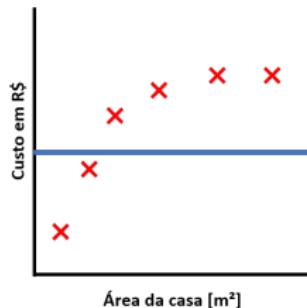
- λ é chamado de **parâmetro de regularização**, e é ≥ 0 .
- Ao escolhermos $\lambda = 0$, eliminamos completamente o efeito da regularização.
- Note que, o primeiro termo da função custo busca adequar o modelo aos dados.
- Enquanto o segundo termo busca manter os parâmetros w_j pequenos.

Implementando a Regularização (caso geral)

Extremos:



$$f(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$



$$f(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

Função custo com regularização

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Perguntas

- Em qual caso acima foi escolhido $\lambda = 0$?
- Em qual caso acima foi escolhido $\lambda = 10^{10}$?

Pergunta:

Considere um algoritmo que inclui um parâmetro de regularização λ . Aumentar λ tenderá a

- A) Aumentar os valores de w_1, w_2, \dots, w_n
- B) Reduzir os valores de w_1, w_2, \dots, w_n
- C) Reduzir o parâmetro b
- D) Aumentar o parâmetro b

Vamos agora resumir como implementar o método do gradiente com regularização tanto para Regressão Linear como também para Regressão Logística

Apenas lembrando que:

- Regressão Linear → Problemas de Regressão (y pode assumir infinitos valores possíveis)
- Regressão Logística → Problemas de Classificação (y assume apenas um pequeno conjunto de valores)

Regressão Linear com Regularização (resumo)

Função custo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Método do Gradiente: repetir

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right) \right]$$

Modelo

$$f_{\vec{w}, b}(\vec{x}^{(i)}) = \vec{w} \cdot \vec{x} + b$$

Tarefa para casa: Deduzir as derivadas.

Regressão Logística com Regularização (resumo)

Função custo:

$$J(\vec{w}, b) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) + (1 - y^{(i)}) \log \left(1 - f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) \right) \right] + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Método do Gradiente: repetir

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^m \left(f_{\vec{w}, b} \left(\vec{x}^{(i)} \right) - y^{(i)} \right) \right]$$

Modelo

$$f_{\vec{w}, b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Tarefa para casa: Deduzir as derivadas.

De olho no código!

De olho no código!

Buscando consolidar nosso conhecimento acerca de regularização, vamos agora implementar novamente o método de regressão logística fazendo as modificações necessárias.

Acesse o Python Notebook usando o QR code ou o link abaixo:

https://colab.research.google.com/github/xaximpv2/master/blob/main/codigo_aula12_Regressao_Logistica_com-Regularizacao.ipynb



Acesse os dados necessários para rodar o código usando o QR code ou o link abaixo:

https://ufprbr0-my.sharepoint.com/:t/g/personal/ricardo_schumacher_ufpr_br/Ee6CfYb1cDFEkmf8FCVXS4B80-1f5UV3dZunU3R_hY-JQ?e=D1WRIf



OBS: Para adicionar os dados ao ambiente do Colab Notebook, no menu do canto esquerdo da tela do Colab clique em "Arquivos" e depois "Fazer upload para o armazenamento da sessão". Então carregue os arquivos baixados.

Parte 1

Rode todo o código. Certifique-se de que você o compreendeu.

Parte 2

- 1 Explique, com suas próprias palavras, o conceito de overfitting e as possibilidades de resolução desse problema.
- 2 Explique, com suas próprias palavras, como implementar a regularização no método do gradiente.
- 3 Qual foi a taxa de acerto (acurácia) obtida com o modelo treinado? Explique, com suas próprias palavras, o que significa essa taxa de acerto. Se necessário, busque pela definição na Internet (ChatGPT, por exemplo).
- 4 Qual seria a taxa de acerto esperada para um modelo com saída 0/1 aleatória?

Parte 3

- 1 Modificando códigos das aulas anteriores, implemente a regularização no contexto da Regressão Linear. Não é necessário entregar esta parte da atividade. Fica apenas como um exercício opcional adicional.