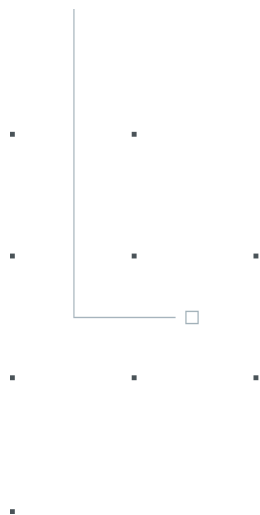


FIAP

NABA



# Classificação de Textos

Prof. Anderson Dourado

- 1. Introdução**
- 2. O Problema de Classificação de Texto**
  1. Extração de características (Vetorização)
  2. Pipeline de transformação
  3. Modelos de classificação
- 3. Análise de sentimentos**
- 4. Métricas de avaliação**
- 5. Amostra**
- 6. Demo**
- 7. Exercícios**

# Introdução



Isto é spam?



Estimado Cliente,

Temos o prazer de informar que finalmente firmamos uma parceria com a Polícia Judiciária em resposta a ataques a sistemas bancários nos últimos anos.

As medidas de segurança do seu cartão MULTIBANCO devem ser atualizadas o mais rápido possível para evitar novos abusos.

**ATUALIZAR AGORA**

Leva apenas 3 minutos,  
Obrigado por nos escolher!

MB WAY  
SIBS

Por favor não responda este email.  
MB WAY  
© 2021 SIBS Payments Solutions.

## Qual a categoria do produto?

1. Kit De Maquiagem Completo Com Necessaire - 18 Itens / MARCAS: MAX LOVE / BELLAFEMME / JASMYNE / SAFIRA / VIVAI / BELLE ANGEL / RUBY ROSE...
2. Harry potter e a ordem da fênix, de Rowling, J. K.. Editora Rocco Ltda, capa mole em português, 2003...

Crítica de filme positiva ou negativa?



Incrivelmente desapontador



Cheio de personagens mirabolantes e uma sátira ricamente aplicada



O melhor filme de comédia já feito



Foi patético. A pior parte foi a cena dentro do saguão.

Texto faz parte de nosso dia-a-dia. Assim, várias aplicações que, de alguma maneira, envolve texto e, principalmente, classificação de texto, surgem constantemente. Podemos elencar alguns exemplos:

- Atribuir categorias, tópicos ou gêneros a assuntos
- Classificação de mensagens textuais
- Identificação autoral
- Identificação de língua escrita
- Classificação de sentimento
- Chatbots.....

Vamos entender, então, o problema de classificação de texto



# O problema de classificação de texto

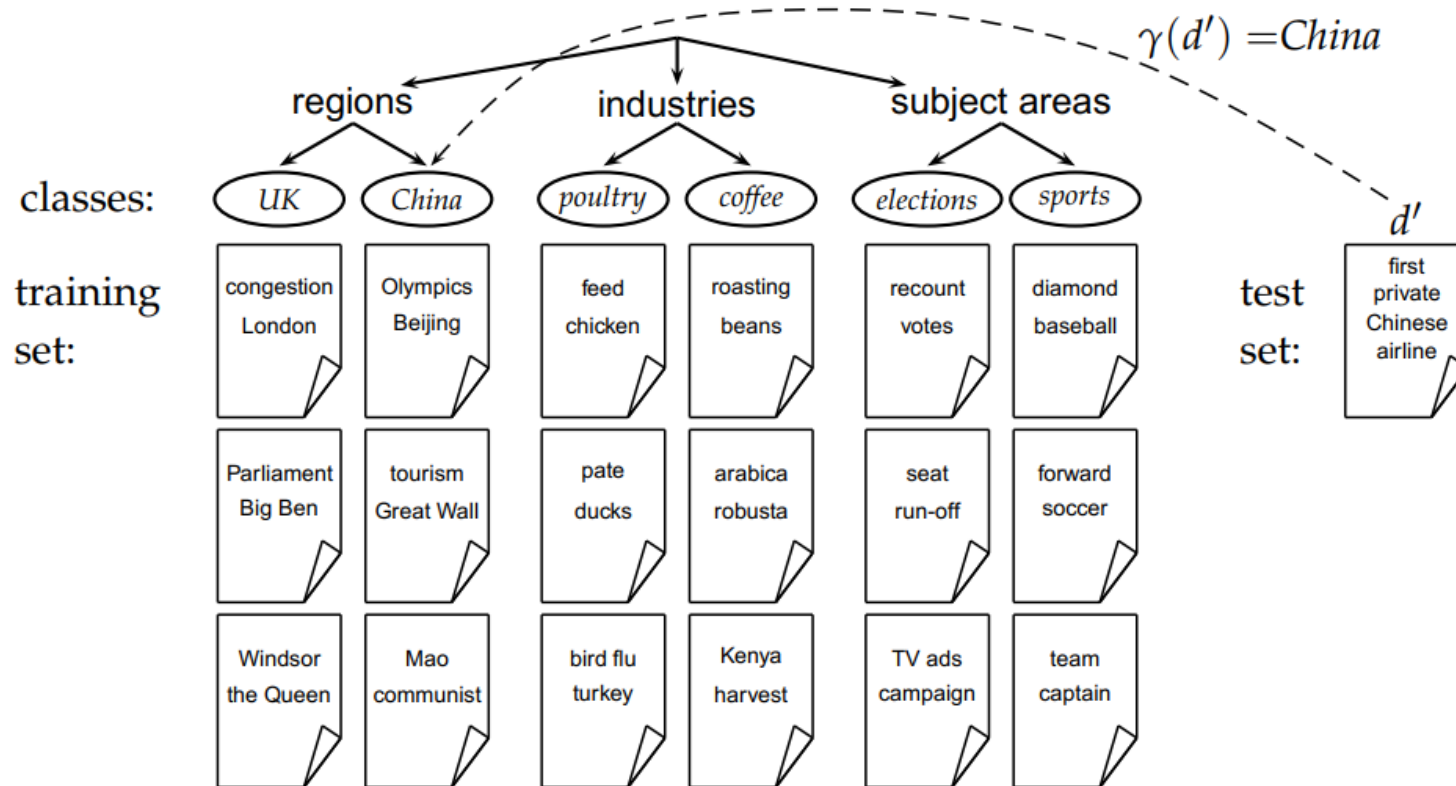
Seja:

- $d \in X$  um documento, em que  $X$  é o espaço de documentos
- $\mathbb{C} = \{c_1, c_2, \dots, c_j\}$  um conjunto fixo de Classes (categorias ou rótulos)
- $\{(d_1, c_1), (d_2, c_2) \dots, (d_m, c_m)\}$  um conjunto de treinamento de  $m$  documentos manualmente rotulados

Usando um [algoritmo de aprendizado](#), desejamos aprender uma função  $\gamma$  de classificação que mapeia documentos para classes:

- $\gamma = X \rightarrow \mathbb{C}$

Podemos representar o problema da seguinte maneira:



Se  $d \in X$  é um documento que pertence ao espaço de documentos, como representa-lo de forma que um computador consiga interpretá-lo?

Como vimos na última aula, é preciso transformar esse documento em números.

O jeito como eu realizo essa transformação é denominado **vetorização**. Inicialmente, veremos três tipos:

- Bag of Words (Contagem simples)
- TF
- TF-IDF

Vamos começar pelo BoW

O modelo Bag of Words usa um vetor de **contagens de palavras** para representar um documento.

$x = [1, 0, 1, 0, 4, 3, 10, 6, 7, \dots]$ , em que  $x_j$  é a contagem da palavra  $j$ .

O tamanho de  $x$  é determinado pelo **vocabulário**  $|\mathcal{V}|$ , que é o conjunto de todas as possíveis palavras no vocabulário

**n-grama + medida = feature**

De modo ilustrado, é assim que funciona o BoW:

I love this movie! It's sweet,  
but with satirical humor. The  
dialogue is great and the  
adventure scenes are fun...  
It manages to be whimsical  
and romantic while laughing  
at the conventions of the  
fairy tale genre. I would  
recommend it to just about  
anyone. I've seen it several  
times, and I'm always happy  
to see it again whenever I  
have a friend who hasn't  
seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

	class	text
0	positivo	Sobre MBA ? Eu gostei muito do MBA da FIAP
1	negativo	O MBA da FIAP pode melhorar, não gostei muito
		0 1
	da	1 1
	do	1 0
	eu	1 0
	fiap	1 1
	gostei	1 1
	mba	2 1
	melhorar	0 1
	muito	1 1
	não	0 1
	pode	0 1
	sobre	1 0

O modelo BoW é assim chamado por somente incluir informação sobre a **contagem de cada palavra**, e **não a ordem em que cada uma aparece**.

Ou seja, com o BoW, a semântica, contexto e as frases em si são ignorados.

Ainda assim, é **surpreendentemente efetivo** para classificação de texto.

Porém podemos ter um problema:

Nem sempre o termo que mais aparece é o mais relevante, por isso temos outras técnicas que derivam desse princípio básico de contagem para começar a extrair sentido do contexto.

Contagem de termos é um valor absoluto e não relativo.



Para entender o conceito de TF e TF-IDF, responda: **todas as palavras num documento são igualmente importantes?**

Diante disso, introduzimos um conceito de frequência de termo que calcula a proporção de um termo num documento em relação ao número total de termos nesse documento.

Entretanto, um problema com pontuar frequências de palavras é que palavras muito frequentes começam a dominar no documento (pontuação alta), mas podem não conter muita “**informação de conteúdo**” para o modelo em relação a palavras raras que pertençam a domínios específicos.

TF é uma contagem relativa dos termos dentro do documento.

$$TF_{w1} = \frac{\text{count}(w1)}{\text{count}(W)}$$

w1 = termo analisado

W = todos os termos do documento

count(w1) = contagem do termos que estamos analisando

count(W) = total de todas os termos individuais sem fazer a distinção

Resolvemos o problema?

class		text	
0	positivo	Sobre MBA ? Eu gostei muito do MBA da FIAP	
1	negativo	O MBA da FIAP pode melhorar, não gostei muito	
		0	1
		da	0.111111 0.125
		do	0.111111 0.000
		eu	0.111111 0.000
		fiap	0.111111 0.125
		gostei	0.111111 0.125
		mba	0.222222 0.125
		melhorar	0.000000 0.125
		muito	0.111111 0.125
		não	0.000000 0.125
		pode	0.000000 0.125
		sobre	0.111111 0.000

Assim, introduzimos um mecanismo para **atenuar o efeito de termos que ocorrem muito nos dados** para tornar significativo a determinação de sua relevância.

Chamamos esse mecanismo de **IDF** (*inverse document frequency*), que **mede a importância de um termo**.

$$IDF_{w_1} = \log_e \left( \frac{\text{count}(D)}{\text{count}(D_{w_1})} \right) \qquad TFIDF_{w_1} = TF_{w_1} * IDF_{w_1}$$

D = todos os documentos

Dw1 = documento que o termo analisado (w1) aparece

count(D) = contagem/total de todos os documentos do corpus

count(Dw1) = contagem dos documento que o termo analisado aparece

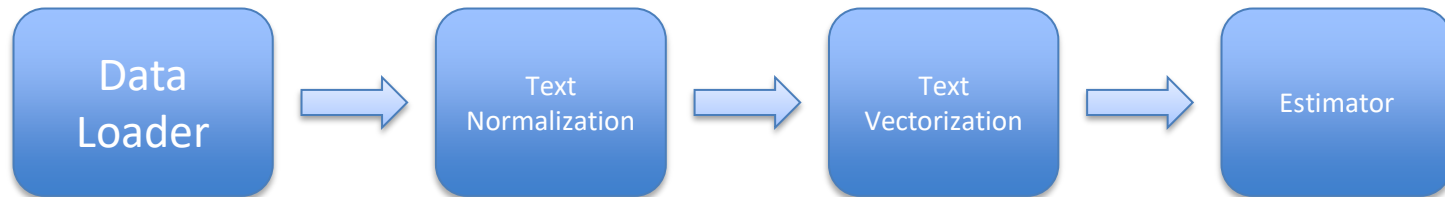
Log\_e = logaritmo natural ou neperiano ( e = 2,718...) – número irracional – número de Euler)

# TF - Term Frequency (Frequência do termo)

class		text					
0	positivo	Sobre MBA ? Eu gostei muito do MBA da FIAP					
1	negativo	O MBA da FIAP pode melhorar, não gostei muito					
words (w1)	contagem		TF (w1/W)		IDF	TF*IDF	
w1	d-1	d-2	d-1	d-2	$\ln(D/Dw1)$	d-1	d-2
da	1	1	0,111	0,125	0	0	0
do	1	0	0,111	0	0,693	0,077	0
eu	1	0	0,111	0	0,693	0,077	0
fiap	1	1	0,111	0,125	0	0	0
gostei	1	1	0,111	0,125	0	0	0
mba	2	1	0,222	0,125	0	0	0
melhorar	0	1	0	0,125	0,693	0	0,087
muito	1	1	0,111	0,125	0	0	0
não	0	1	0	0,125	0,693	0	0,087
pode	0	1	0	0,125	0,693	0	0,087
sobre	1	0	0,111	0	0,693	0,077	0
Total (W)	9	8					

Até agora, vimos uma série de transformações que podemos fazer nos dados textuais, mas como organizar isso de modo a construir um pipeline a fim de realizar essas transformações?

Um simples pipeline pode consistir das seguintes etapas:



O processo de normalização de texto pode ser composto pelas seguintes partes:

- Tokenização
- Remoção de stop-words
- Lematização e/ou Stemização
- Remoção de pontuação
- Lowercasing
- Entre outros tratamentos...

Na demo, eu apresento o processo completo.

Uma vez que **geramos as features** aplicando uma das estratégias de **vetorização**, fica simples **treinar um modelo de machine learning** para, por exemplo, classificar novos documentos.

Treinando um **modelo n-grama** de árvore de decisão com base no texto previamente vetorizado:

```
from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier()
tree.fit(text_vect, df['class'])

print('D Tree: ', tree.score(text_vect, df['class']))
```

Preparando dados e fazendo predição

```
texto = vect.transform(['a curso pode melhorar'])

print('D Tree: ', tree.predict(texto))
```

# Análise de Sentimentos



A análise de sentimentos nada mais é que uma forma de classificação de texto. Vimos esses exemplos anteriormente sobre avaliação de filmes. Nesse caso, o objetivo da classificação de texto consiste em, dado um novo review, predizer seu sentimento.



Incrivelmente desapontador



Cheio de personagens mirabolantes e uma sátira ricamente aplicada



O melhor filme de comédia já feito



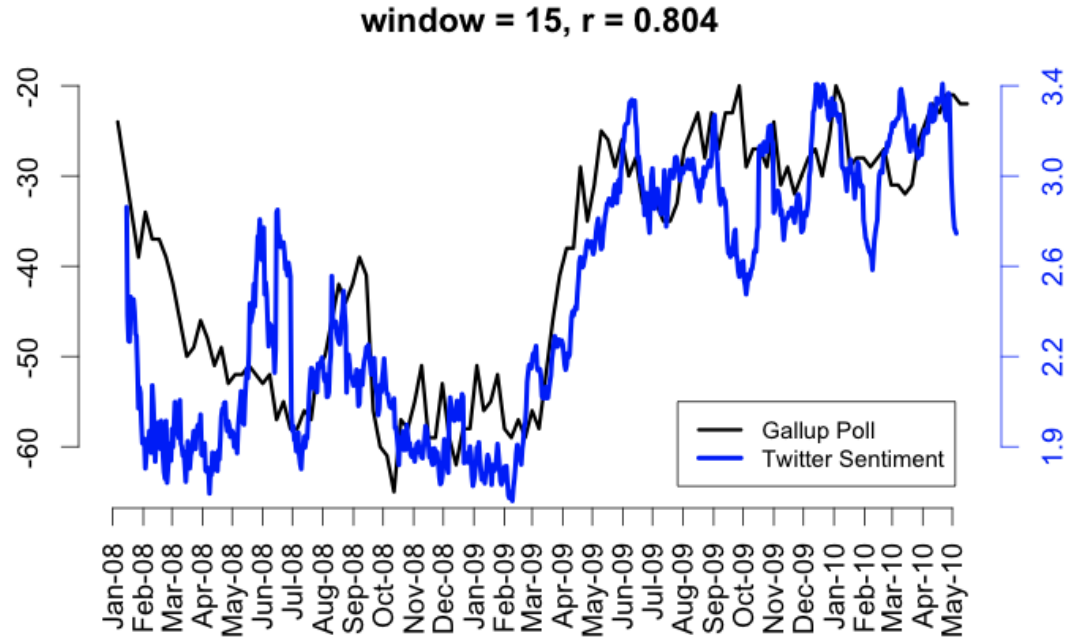
Foi patético. A pior parte foi a cena dentro do saguão.

E por qual razão Análise de Sentimentos é importante?

Bem, ela faz parte de nosso dia-a-dia:

- **Filmes**: o que a crítica e o público tem dito acerca do novo lançamento?
- **Produtos**: o que as pessoas pensam sobre o novo Iphone?
- **Sentimento público**: como está a confiança do consumidor?
- **Política**: o que as pessoas pensam a respeito de um candidato?
- **Predição**: qual a tendencia de mercado?

Esse gráfico compara o sentimento extraído de tweets com a pesquisa de confiança do consumidor realizada pela Gallup Poll:



# Métricas de Avaliação



You can't  
manage what  
you can't  
measure

-Peter Drucker

Precisamos de maneiras de mensurar a qualidade de predição de nosso algoritmo. Em classificação, as métricas mais comuns, e as mais simples, são **Acurácia**, **Precision**, **Recall** e **F1 Score**. Elas são obtidas a partir da [Matriz de Confusão](#), apresentada abaixo:

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

A acurácia é dada pela seguinte função:

$$acc = \frac{tp + tn}{(tp + tn + fp + fn)}$$

Precision (acurácia das predições positivas) pode ser calculado da seguinte maneira:

$$P = \frac{tp}{(tp + fp)}$$

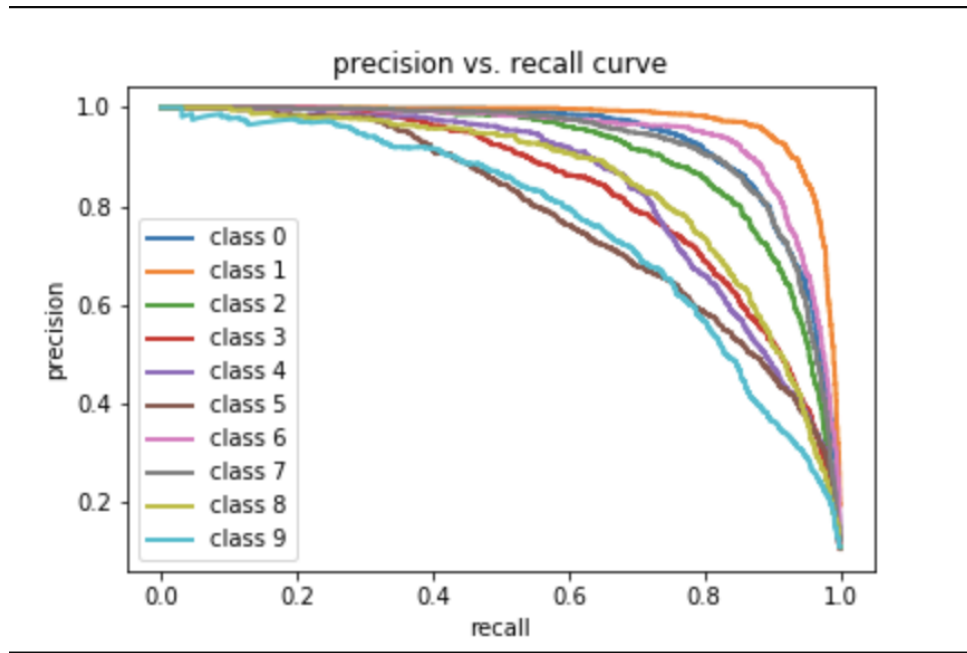
Já o Recall (taxa de amostras positivas corretamente encontradas pelo classificador) é calculado da seguinte forma:

$$R = \frac{tp}{(tp + fn)}$$

**Precision** e **Recall** tendem a ser inversamente proporcionais, como apresentado no gráfico ao lado.

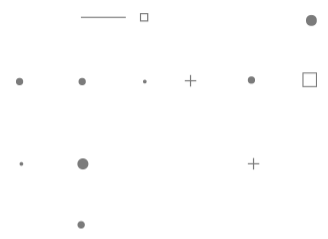
Para minimizar esse problema, existe a métrica **F1-Score**, que leva em consideração tanto Precision quanto Recall.

$$F1 = \frac{2 * P * R}{P + R}$$





# Amostra

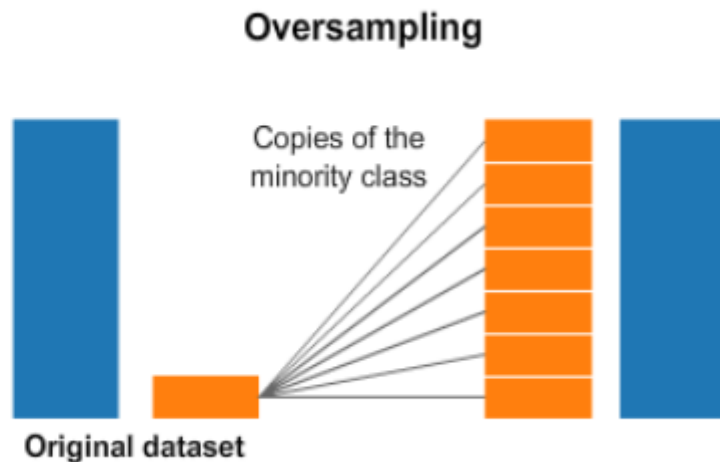
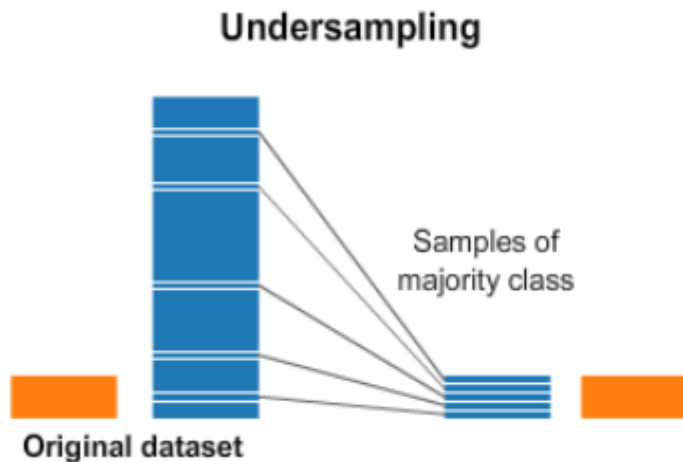


Selecionar um grupo de dados que represente o total dos dados.

Método: COLETA > ANÁLISE > CONCLUSÃO

## Podemos trabalhar com todos os dados?

Independente disso temos que pensar na amostra que vamos usar para teste e/ou validação do nosso modelo.



Existem muitas formas de desenvolver uma amostra.

Exemplo de uma formula comum usada para calcular uma amostragem que pode representar uma determinada população para uma pesquisa ou análise.

$$\alpha = 1 - \gamma$$

$$n0 = 1 / \alpha^2$$

$$n = N * n0 / N + n0$$

N = Tamanho total da população (dados);

n0 = Tamanho inicial da Amostra com um erro como parâmetro;

n = Tamanho da Amostra para representar o total dentro do erro esperado;

$\gamma$  = Grau de confiança esperado da amostra em relação a população;

$\alpha$  = Probabilidade de erro esperado.

Total população (N)	Amostra (n)	% do total
500	222	44,4%
1.000	286	28,6%
2.000	333	16,7%
3.000	353	11,8%
10.000	385	3,8%
100.000	398	0,4%
1.000.000	400	0,04%

Grau de confiança de 95%

## Atenção!

- Uma amostra sempre deve ser aleatória dentro do universo escolhido.
- Não é consenso as formas de estratificar uma amostra.
- Toda amostra envolve erros atrelados.
- Use o bom senso, testando e validando sua amostra (amostra heterogênea).
- Se pode usar toda a massa de dados, essa pode ser a “melhor amostra”.

# Demo e Exercícios



## Exercício - Vamos resolver um problema?

Uma empresa de marketplace, disponibiliza sua plataforma de para diversos vendedores cadastrarem seus produtos em diferentes categorias previamente definidas. Essas categorias são utilizadas para melhor distribuir a divulgar seus produtos para os clientes e usuários da plataforma.

Mas nem todos os vendedores respeitam essas categorias, regras e as diretrizes do marketplace, pense nos diversos problemas que podemos enfrentar:

- Vendedores que cadastram produtos em categorias erradas;
- Vendedores que querem vendem produtos que não são permitidos pelas políticas do marketplace e por ai vai...

**Será que é possível validar produto por produto? um por um? ...que trampo!!!**

Você Cientista de Dados, consegue ajudar a mitigar esse problema? Conseguiria criar algum mecanismo de diminua esse trabalho manual?

Bom, podemos criar um **modelo que seja capaz de classificar um produto** através do nome e da descrição, e depois podemos confrontar com a categoria e premissas da plataforma.

O primeiro passo que podemos “dar” é através de uma base de dados de produtos categorizados treinar um modelo de classificação. **Vamos começar explorando essa base de dados?**

Como extrair valor para uma empresa a partir dos dados?  
Exemplo de um **Ciclo Analítico** para processo de modelagem.

## Aquisição de dados e análise exploratória

Preparar uma base de dados com as variáveis elegíveis para resolução do problema.  
Em seguida: análise exploratória, limpeza, transformação e seleção das melhores variáveis para criar o modelo.

## Modelo

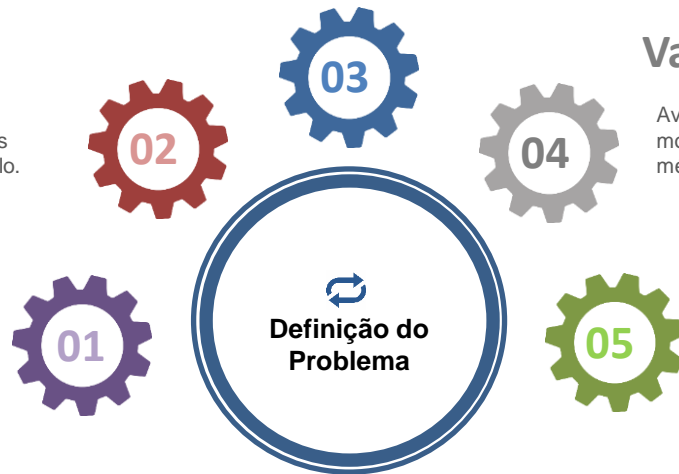
Tudo certo? É hora de escolher os algoritmos para fazer a tarefa.

## Validação

Avaliar os resultados do modelo por meio de métricas.

## Planejamento

Definir a variável resposta e a estratégia de treino do modelo. Capturar também todas as informações possíveis como expectativas das pessoas, fontes de dados, variáveis iniciais, modelos parecidos...



## Produção

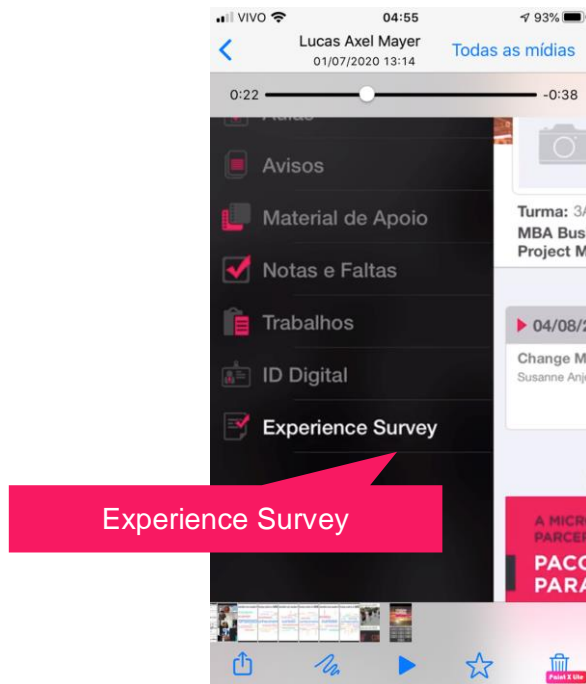
Pensar como o modelo será operacionalizado e colocá-lo em produção.

**Ciclo de vida do modelo**

O que acharam da aula?


Pelo aplicativo da FIAP ou pelo site

(Entrar no FIAP, e no menu clicar em Experience Survey)



# Obrigado!

profandeson.dourado@fiap.com.br

 /anderson-dourado

FIAP MBA<sup>+</sup>

Copyright © 2023 | Professor Anderson Vieira Dourado  
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.



FIAP