Nanodegree Engenheiro de Machine Learning

1 – Definição

Visão geral do Projeto

Ao longo desses anos o telemarketing sofreu uma série de mudanças, ou melhor, transformações que estão relacionadas diretamente com a evolução da tecnologia. Surgiram conceitos como Omnichannel (multicanal), Callback, Atendentes Virtuais, Chatbot, Portais de Autoatendimento e etc.

O cliente mudou, atualmente ele está no controle, não quer esperar 'pendurado' na linha e também não quer ser importunado por ligações oferecendo produtos que não são do seu interesse. Ele pode a qualquer momento deixar de comprar na sua loja, por exemplo, e escolher o mesmo produto em outra loja utilizando qualquer meio (ex: smartphone, tablet). Por outro lado, existe um potencial público de clientes que pode contratar o produto, desde que, seja realizado um contato e ofertado o produto adequado.

Nesse cenário é que o projeto se enquadra, utilizando machine learning para selecionar o público que será ofertada a Campanha. O modelo de machine learning treinado e testado conseguirá prever quais são os clientes mais propensos a contratar um empréstimo bancário.

O conjunto de dados utilizado para treinar o modelo é referente a uma campanha de marketing realizada por uma instituição bancária portuguesa. Essa campanha foi baseada em chamadas telefonicas. O produto ofertado foi um empréstimo a prazo.

Esse conjunto de dados é público, disponivel para pesquisa. Os detalhes estão descritos em [Moro et al., 2014]. O conjunto foi obtido no site da UCI (https://archive.ics.uci.edu/ml/datasets/bank+marketing)

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

A solução para esse problema é a execução de um modelo de aprendizagem supervisionado com esse conjunto de dados. Será necessário efetuar uma um

processo de preparação dos dados. Esse modelo será treinado com esse conjunto (já preparado), gerando no final do seu processamento, um modelo matemático que consegue prever (a partir dos dados do cliente) se o cliente irá contratar o produto/serviço bancário.

Será feito um split do conjunto de dados, usando 80% para aprendizagem (treinamento) e 20% para testes. Os resultados dos testes serão comparados com os dados reais e dessa forma será possível medir a efetividade do modelo gerado.

A partir desse momento, nas próximas campanhas, a seleção do público será feita apresentando o novo conjunto para o modelo já treinado. Ao término da execução do modelo teremos a informação de quais clientes provavelmente contratarão o produto.

Descrição do Problema

O problema que estamos propondo solucionar é selecionar de forma mais assertiva o público que será ofertada campanhas de marketing do produto empréstimo a prazo de uma instituição bancária. A seleção não assertiva implica um alto gasto (estudo e seleção do público, de tempo, em pessoas, em telefonia, etc) contatando clientes que não tem interesse em contratar o produto.

Dessa formar será dado um foco nos clientes que realmente desejam contratar o produto, com isso a instituição financeira poderá investir mais 'energia' em melhor atender e oferecer serviços mais atraentes aos seus clientes.

Para resolver esse problema trabalhei com um conjunto de dados que possui as informações os clientes e também a feature que informa se o cliente contratou ou não o produto. Face esse cenário utilizei um modelo supervisionado de classificação, pois, nosso problema requer um modelo que retorne uma classificação, no caso, classifique os clientes como contratará ou não contratará o produto ofertado. Com essa resposta, podemos em contatar os clientes que o modelo classificou como 'sim', ou seja, contratará o produto.

O conjunto de dados possui tanto variáveis numéricas como variáveis categóricas. Utilizei ambos os tipos, porém, para as variáveis categóricas foi necessário transformá-las em numéricas para que o modelo possa funcionar.

Métricas

As métricas utilizadas serão (do Pacote Scikit.metrics):

accuracy_score (acurácia): Proporção de casos que foram corretamente previstos, sejam eles verdadeiro positivo ou verdadeiro negativo. Com isso, mede a frequência que o modelo faz a previsão correta.

f1_score (score): Média ponderada dos scores de precisão (precision) e sensibilidade (recall). Varia entre 0 e 1, sendo 1 a melhor pontuação possível.

precision_score (precisão): Proporção de casos positivos que o modelo classificou como positivo e eram positivas, ou seja, é a capacidade de não rotular como positiva uma amostra que é negativa (Verdadeiro Positivo / (Verdadeiro Positivo + Falso Positivo))

recall_score (sensibilidade): Proporção de casos realmente positivos e que foram classificados pelo modelo como positivo, ou seja, é a capacidade do modelo de encontrar todas as amostras positivas. ((Verdadeiro Positivo / (Verdadeiro Positivo + Falso Negativo))

- confusion_matrix (Matriz de confusão): Essa métrica calcula a quantidade de falso positivo e falso negativo, e de verdadeiro positivo e verdadeiro negativo.

A principal métrica que foi utilizada para decidir qual modelo será escolhido foi a acurácia, pois, ela mede o quanto o modelo faz a previsão correta. Como o problema a ser resolvido é prever quais clientes contratarão o produto, a métrica acurácia é a que mais se enquandra.

2 – Análise

Exploração de Dados

Primeiramente, segue abaixo as informações de cada feature do conjunto:

ID	Label	Descrição	Tipo	Dominio
1	Age	Idade	Numérica	
2	Job	Tipo de Emprego	Categórica	admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown'

3	Marital	Estado Civil	Categórica	divorced','married','single','unknown'; note: 'divorced' means divorced or widowed
4	Education	Escolaridade	Categórica	basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown'
5	Default	Tem Crédito Default	Categórica	no','yes','unknown'
6	Housing	Tem Empréstimo Habitacional	Categórica	no','yes','unknown'
7	Loan	Tem Empréstimo Pessoal	Categórica	no','yes','unknown'
8	Contact	Tipo de Contato	Categórica	cellular','telephone'
9	Month	Mês do último contato	Categórica	jan', 'feb', 'mar', ..., 'nov', 'dec'
10	day_of_week	Último dia de contato da semana	Categórica	'mon','tue','wed','thu','fri'
11	Duration	Duração do último contato em segundos	Numérica	Nota importante: este atributo afeta altamente a meta de saída (por exemplo, se a duração for = 0, então y = 'não'). No entanto, a duração não é conhecida antes de uma chamada ser executada. Além disso, após o término da chamada, é obviamente conhecido. Assim, essa entrada deve ser incluída apenas para fins de benchmark e deve ser descartada se a intenção for ter um modelo preditivo realista.
12	campaign	Número de contatos da última camapanha	Numérica	
13	Pdays	Número de dias que se passaram depois que o cliente foi contatado pela última vez de uma campanha anterior. 999 indica que cliente não foi contato em campanha anterior	Numérica	
14	Previous	Número de contatos realizados antes desta campanha	Numérica	
15	poutcome	Resultado da campanha de marketing anterior	Categorica	failure','nonexistent','success'
16	emp.var.rate	Taxa de variação de emprego	Numérica	
17	cons.price.idx	Indice de preços ao consumidor	Numérica	
18	cons.conf.idx	Indice de confiança do consumidor	Numérica	
19	euribor3m	Euro	Numérica	
20	nr.employed	Número de empregos	Numérica	
21	Target	Cliente contratou produto bancário	Binaria	yes','no'

Para ilustrar, segue duas amostras do conjunto, cada uma com 10 registros:

10 casos que não contrataram os produtos

age	job	Marital	Education	default	housing	loan	contact	month
56	housemaid	Married	basic.4y	no	no	no	telephone	may
57	services	Married	high.school	unknown	no	no	telephone	may
37	services	Married	high.school	no	yes	no	telephone	may
40	admin.	Married	basic.6y	no	no	no	telephone	may
56	services	Married	high.school	no	no	yes	telephone	may
45	services	Married	basic.9y	unknown	no	no	telephone	may
59	admin.	Married	professional.course	no	no	no	telephone	may
41	blue-collar	Married	Unknown	unknown	no	no	telephone	may
24	technician	Single	professional.course	no	yes	no	telephone	may
25	services	Single	high.school	no	yes	no	telephone	may

day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate
mon	261	1	999	0	nonexistent	1.1
mon	149	1	999	0	nonexistent	1.1
mon	226	1	999	0	nonexistent	1.1
mon	151	1	999	0	nonexistent	1.1
mon	307	1	999	0	nonexistent	1.1
mon	198	1	999	0	nonexistent	1.1
mon	139	1	999	0	nonexistent	1.1
mon	217	1	999	0	nonexistent	1.1
mon	380	1	999	0	nonexistent	1.1
mon	50	1	999	0	nonexistent	1.1

cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no
93.994	-36.4	4.857	5191	no

10 casos que contrataram produtos

age	job	Marital	Education	default	housing	loan	contact	month
41	blue-collar	Divorced	basic.4y	unknown	yes	no	telephone	may

49	entrepreneur	Married	university.degree	unknown	yes	no	telephone	may
49	technician	Married	basic.9y	no	no	no	telephone	may
41	technician	Married	professional.course	unknown	yes	no	telephone	may
45	blue-collar	Married	basic.9y	unknown	yes	no	telephone	may
42	blue-collar	Married	basic.9y	no	yes	yes	telephone	may
39	housemaid	Married	basic.9y	no	yes	no	telephone	may
28	unknown	Single	Unknown	unknown	yes	yes	telephone	may
44	services	Married	high.school	no	yes	no	telephone	may
42	technician	Married	professional.course	no	no	no	telephone	may

day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate
mon	1575	1	999	0	nonexistent	1.1
mon	1042	1	999	0	nonexistent	1.1
mon	1467	1	999	0	nonexistent	1.1
mon	579	1	999	0	nonexistent	1.1
mon	461	1	999	0	nonexistent	1.1
mon	673	2	999	0	nonexistent	1.1
mon	935	3	999	0	nonexistent	1.1
Tue	1201	1	999	0	nonexistent	1.1
Tue	1030	1	999	0	nonexistent	1.1
Tue	1623	1	999	0	nonexistent	1.1

cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
93.994	-36.4	4.857	5191	yes
93.994	-36.4	4.857	5191	yes
93.994	-36.4	4.857	5191	yes
93.994	-36.4	4.857	5191	yes
93.994	-36.4	4.857	5191	yes
93.994	-36.4	4.857	5191	Yes
93.994	-36.4	4.857	5191	Yes
93.994	-36.4	4.857	5191	Yes
93.994	-36.4	4.857	5191	Yes
93.994	-36.4	4.857	5191	Yes

Foram realizadas as seguintes análises nas features:

- Identificação de features com ausência de valores: Nenhuma feature tem valor ausente.

- Identificação de registros duplicados. Foi considerado duplicado, caso todas as features estejam com o mesmo conteúdo de outro registro

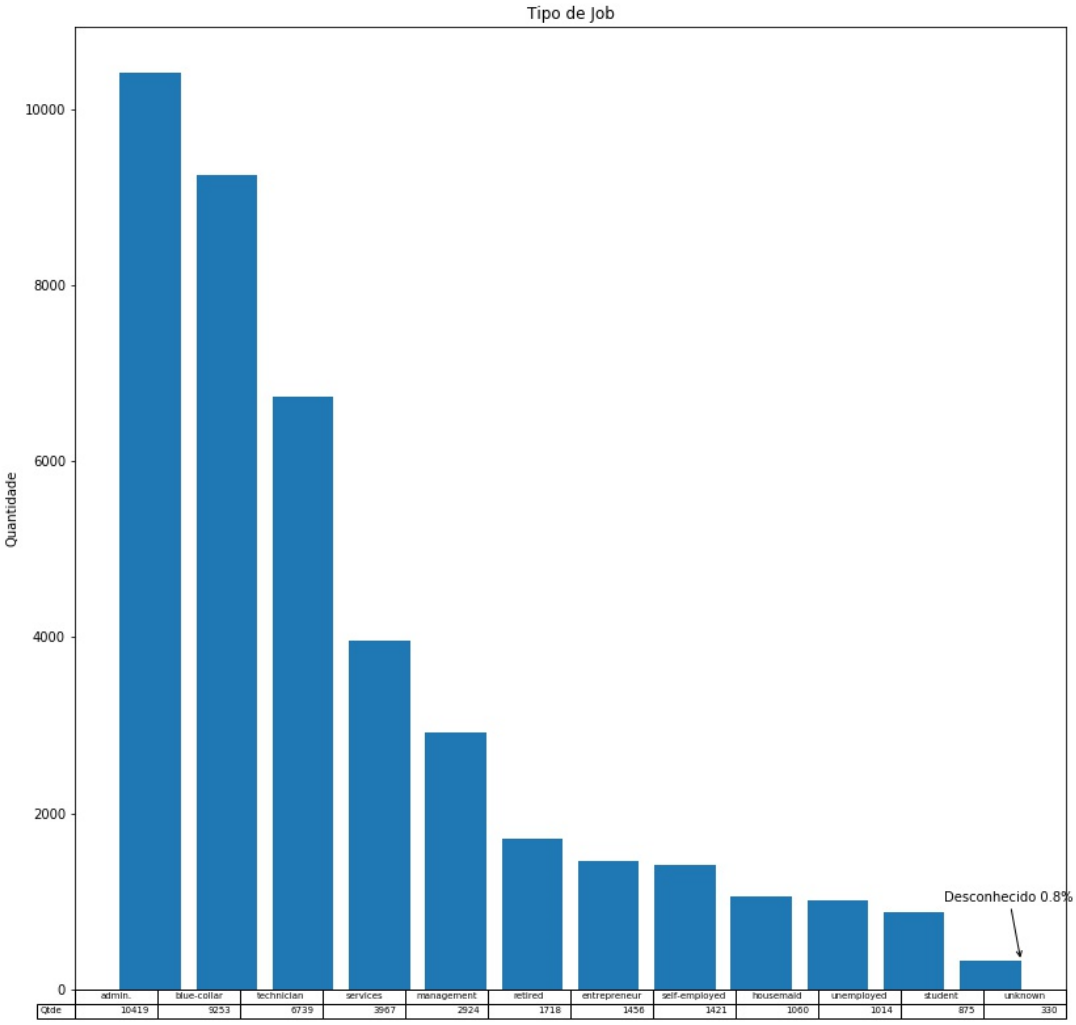
- Para as features numéricas foi visualizado e analisado a média, mediana, desvio padrão e máximo valor. Não foi identificado nenhuma grande variação. Um ponto importante identificado foi que as features possuem escalas de valores variadas

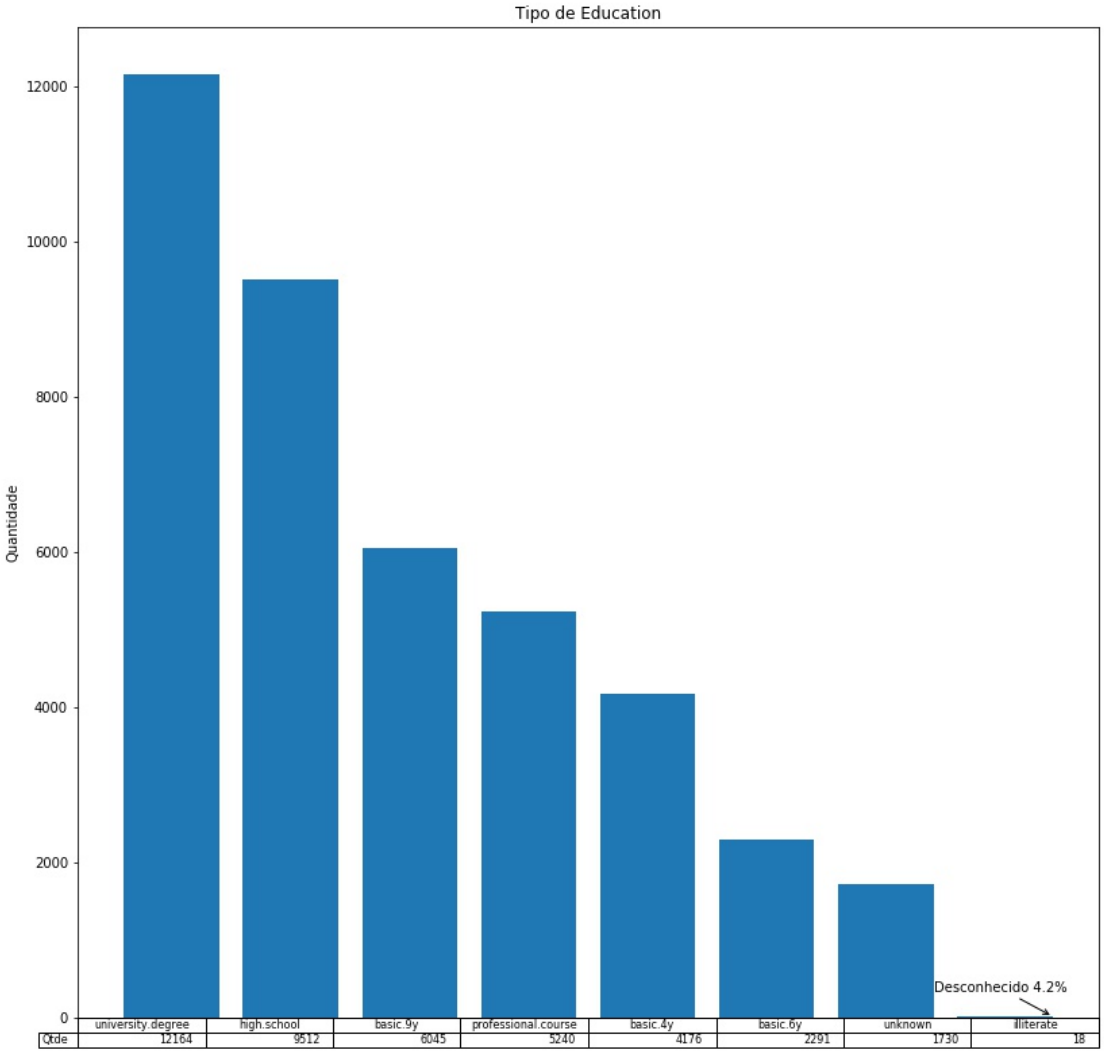
```
dataset.describe()
```

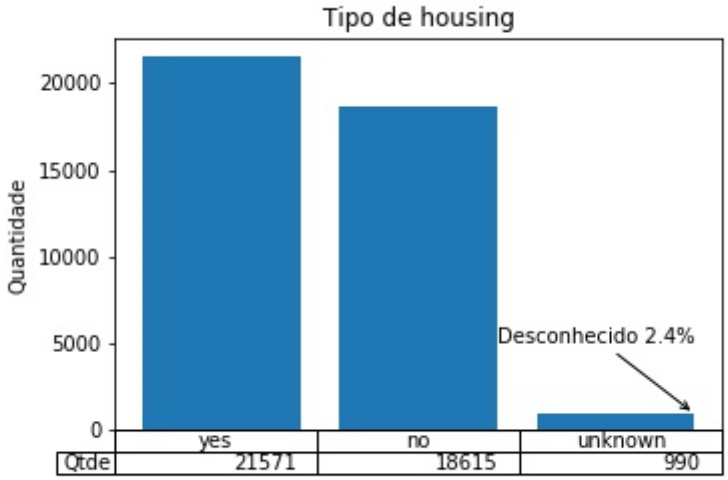
	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41176.00000	41176.000000	41176.000000	41176.000000	41176.000000	41176.000000	41176.000000	41176.000000	41176.000000	41176.000000
mean	40.02380	258.315815	2.567879	962.464810	0.173013	0.081922	93.575720	-40.502863	3.621293	5167.034870
std	10.42068	259.305321	2.770318	186.937102	0.494964	1.570883	0.578839	4.627860	1.734437	72.251364
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

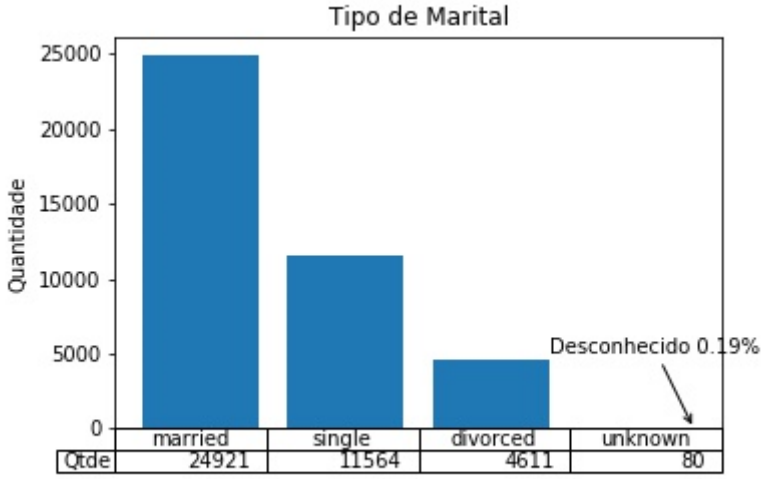
feature	mediana	media
age	38	40,02380027
campaign	2	2,567879347
pdays	999	962,4648096
previous	0	0,173013406
emp.var.rate	1,1	0,081921508
cons.price.idx	93,749	93,57571989
cons.conf.idx	-41,8	-40,50286332
euribor3m	4,857	3,621293448
nr.employed	5191	5167,03487

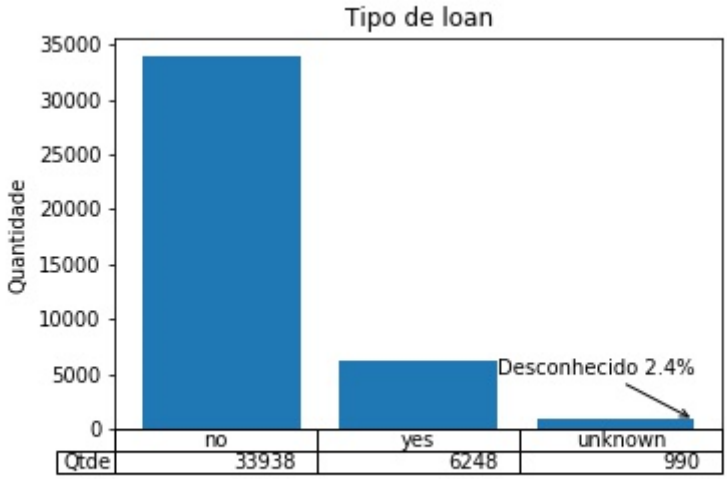
- Para as variáveis categóricas a visualização foi realizada através de um agrupamento para obter a quantidade (por domínio) de cada feature.
    - Foi constatado que a feature 'default' possui muitos registros 'sem informação', ou seja, é uma variável de pouca representatividade
    - A feature 'duration' é obtida somente após a campanha ser realizada, ou seja, não pode ser utilizada para realizar previsões
    - A feature 'housing' não possui um valor dominante.
    - A maior parte das features categóricas possuem registros com registros com valores 'unknown', porém, não são valores dominantes. Para representar, os resultados foram plotados em gráficos (por feature)


Tipo de Job
Quantidade
10000
8000
6000
4000
2000
0
Desconhecido 0.8%
admin. blue-collar technician services management retired entrepreneur self-employed housemaid unemployed student unknown
Qtde 10419 9253 6739 3967 2924 1718 1456 1421 1060 1014 875 330

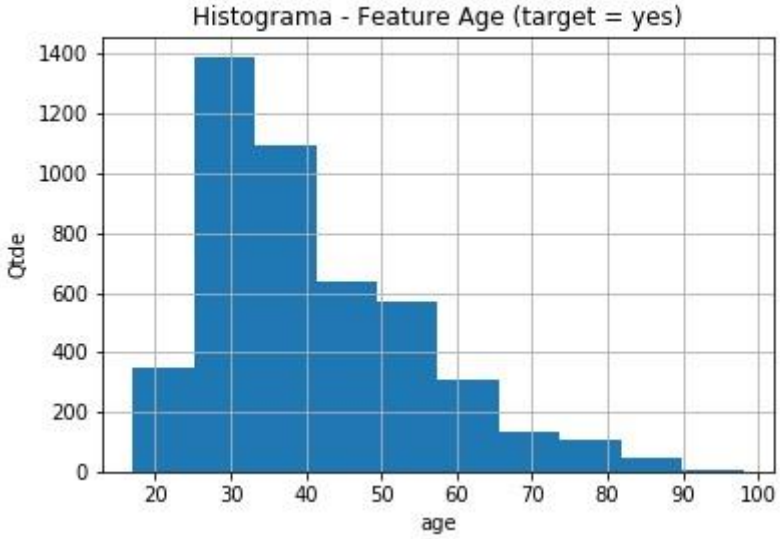

Tipo de Education
Quantidade
12000
10000
8000
6000
4000
2000
0
Desconhecido 4.2%
university.degree
high.school
basic.9y
professional.course
basic.4y
basic.6y
unknown
illiterate
Qtde
12164
9512
6045
5240
4176
2291
1730
18


Tipo de housing
Quantidade
20000
15000
10000
5000
0
Desconhecido 2.4%
yes
no
unknown
Qtde
21571
18615
990

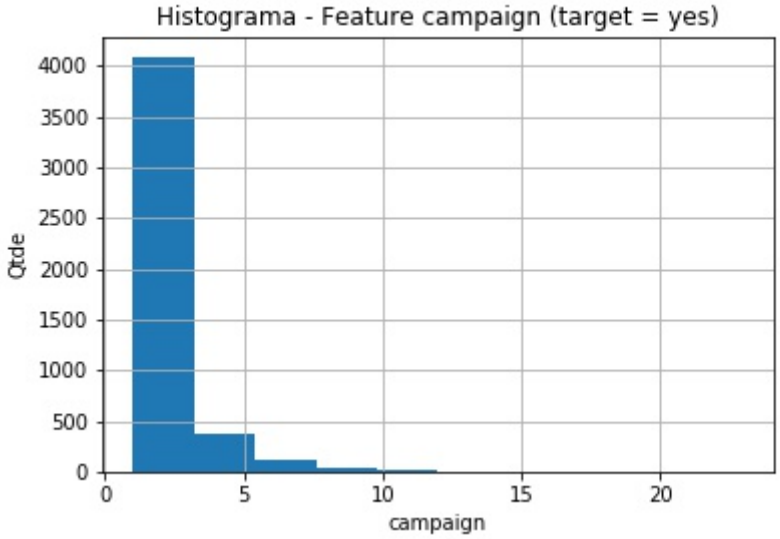




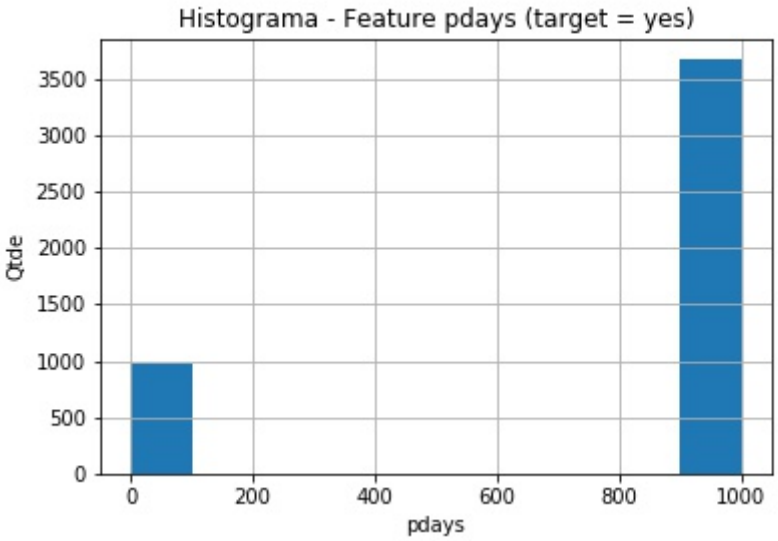
- Foi utilizado um método para identificar valores discrepantes (outliers)
- Analisei também os clientes que contrataram o produto (y=yes):



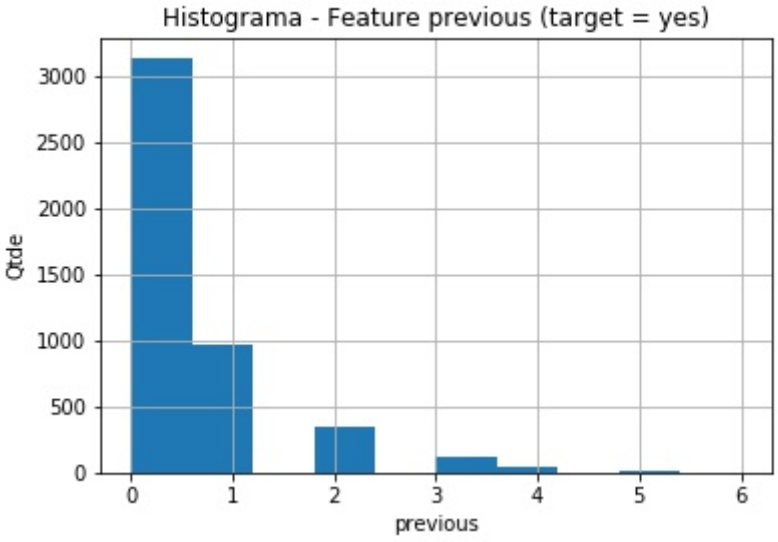
O histograma da feature age mostrou que entre 25 e 35 são os clientes que mais contratam o produto.



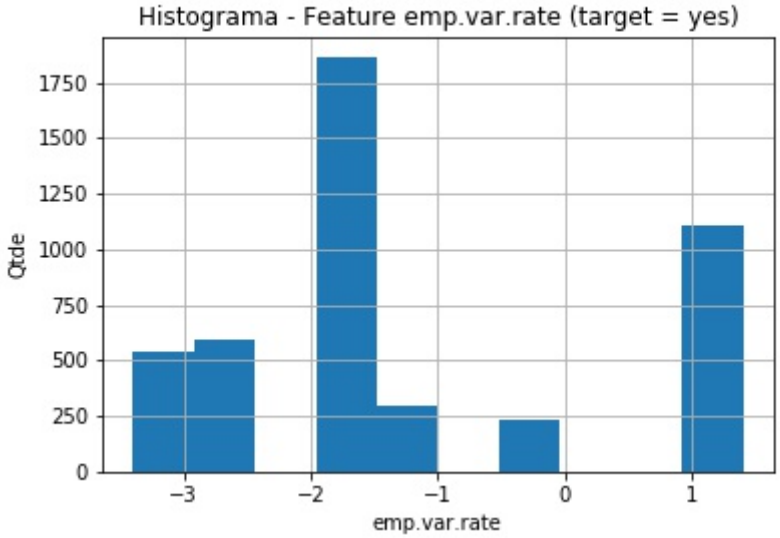
O histograma da feature campaign mostrou que praticamente todos os clientes que contrataram o produto somente foram contatados em uma (1) campanha.



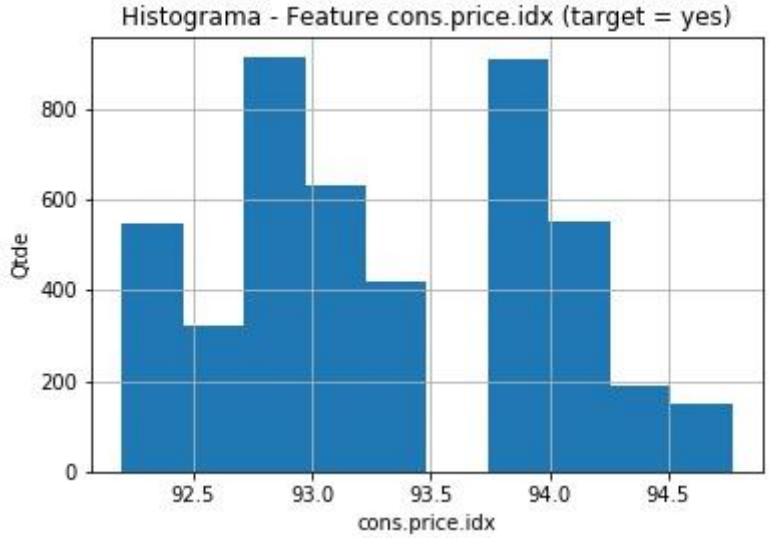
O histograma da feature pdays mostrou que praticamente todos os clientes que contrataram o produto não foram acionados por outras campanhas (pdays=999)



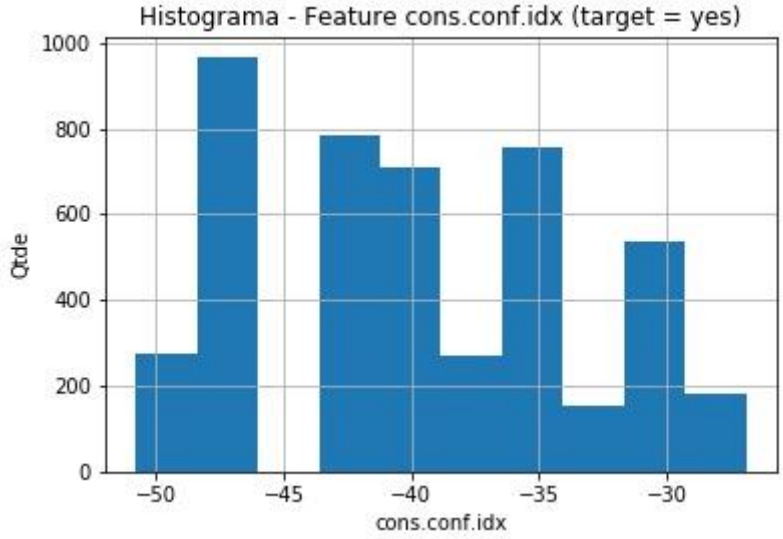
O histograma da feature previous mostrou que praticamente todos os clientes possuem previous = 0, ou seja, não tiveram nenhum contato antes dessa campanha



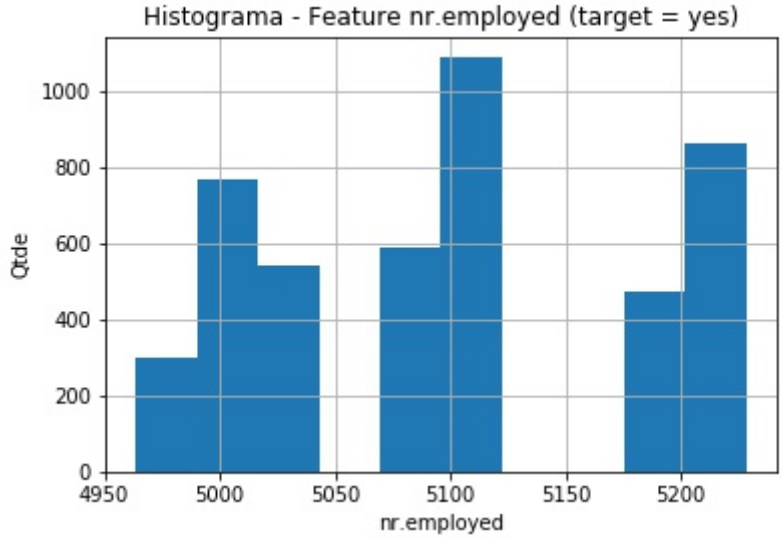
O histograma da feature emp.var.rate mostrou uma tendência maior para os clientes entre -2 e 1,5



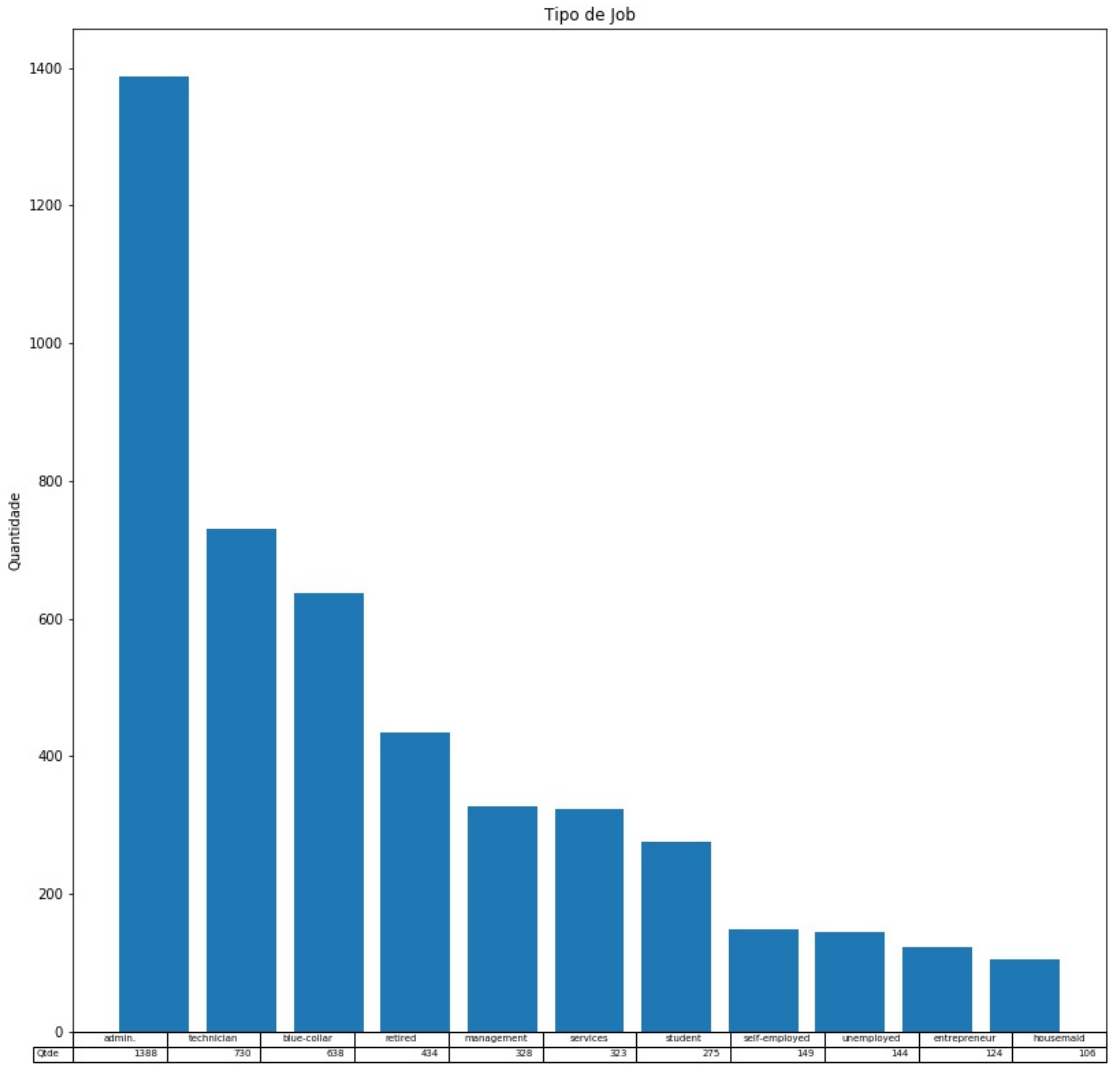
O histograma da feature cons.price.idx não mostrou uma tendência clara entre os valores



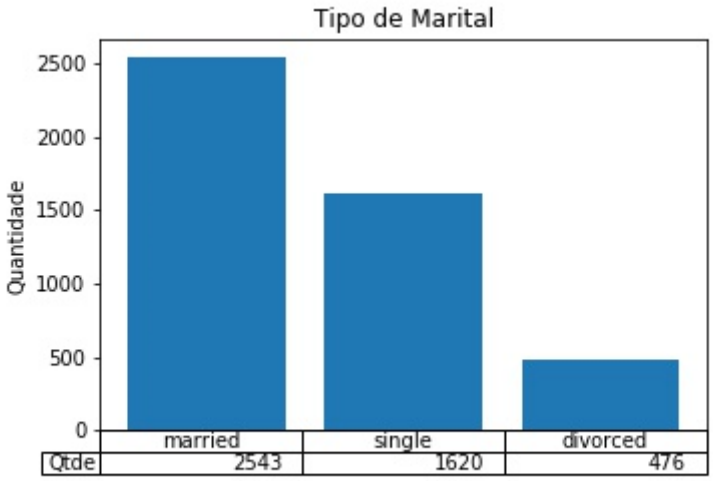
O histograma da feature cons.conf.idx não mostrou uma tendência clara entre os valores



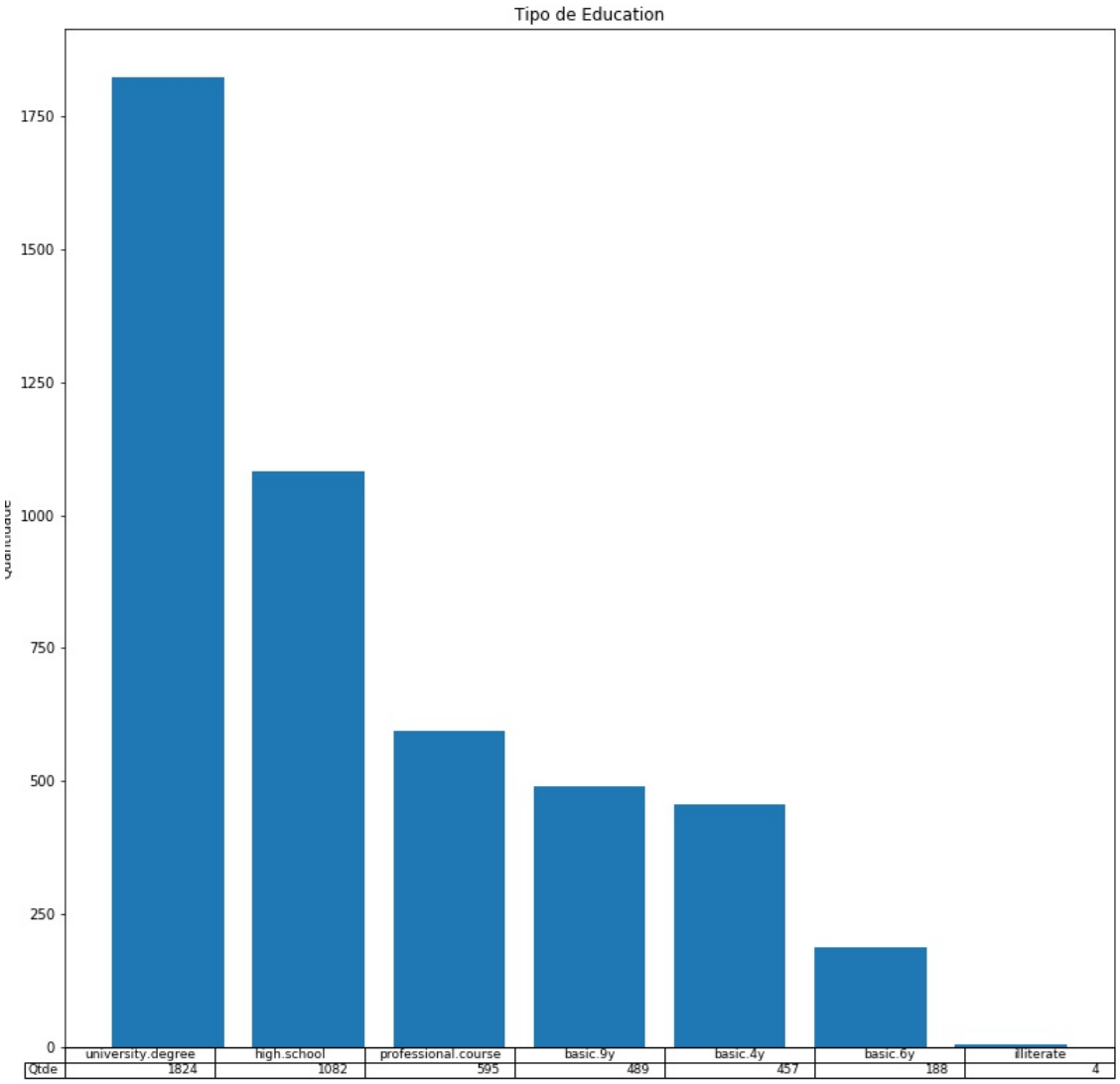
O histograma da feature nr.employed não mostrou uma tendência clara entre os valores



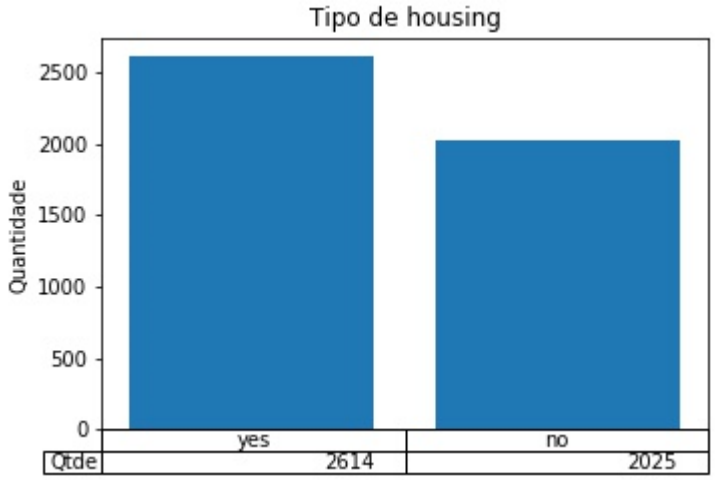
O gráfico da feature job mostrou um número maior de clientes com job = admin.



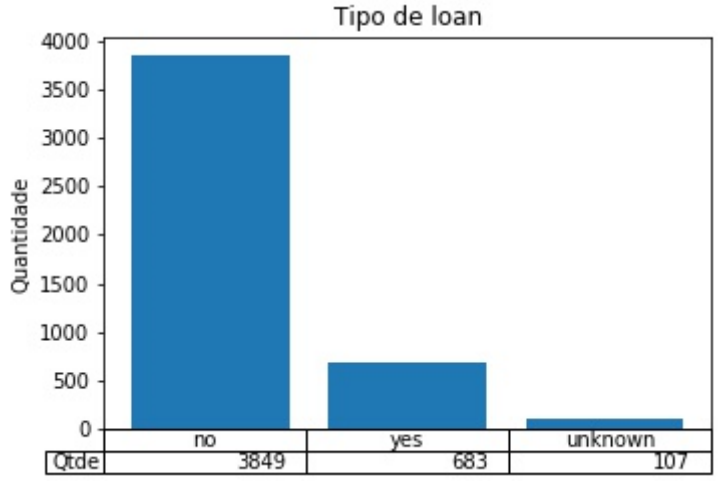
O gráfico da feature marital mostrou uma tendência maior dos clientes com a feature marital=married.



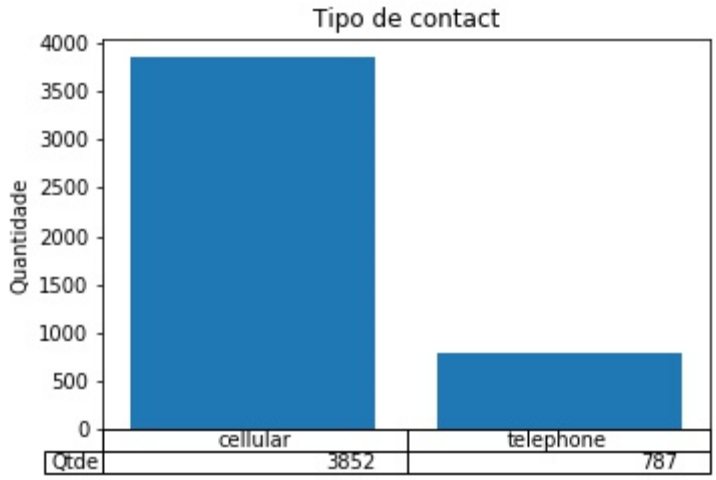
O gráfico da feature education mostrou um número grande de clientes com education=university.degree. Para as outros tipos de formação, que provavelmente possuem uma renda menor, poderiam ser ofertadas opções diferentes (por exemplo, com prazo maior e/ou parcelas menores) ou mesmo outro tipos de produtos bancários



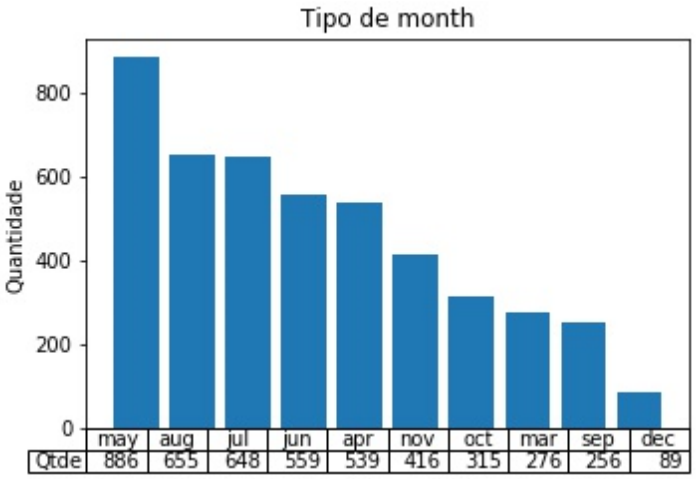
O gráfico da feature housing não mostrou uma tendência clara dos clientes que contrataram o produto.



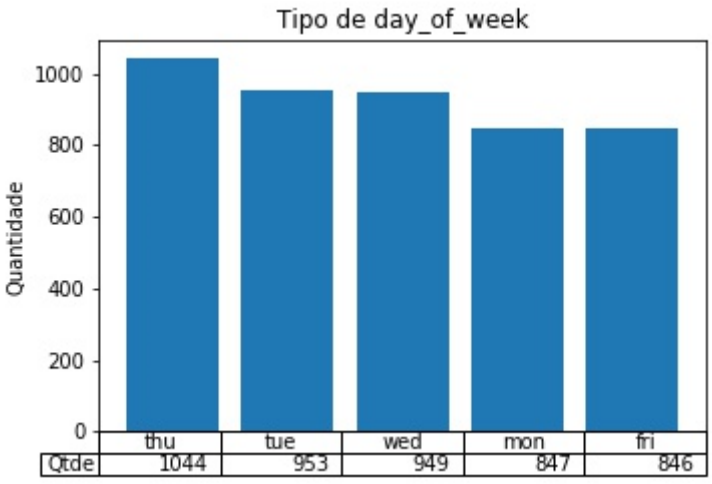
O gráfico da feature loan mostrou que a grande maioria dos clientes não possuem loan (no). Isso pode significar que clientes que já possuem empréstimo (loan) não devem ser alvos de novas companhas ou que para esse público deve ser ofertado um outro tipo de produto ou opções diferenciadas do mesmo produto.



O gráfico da feature contact mostrou que a grande maioria dos clientes que contrataram o produto foram contatados por cellular. Vale analisar se a baixa adesão nos contatos via telefone não está relacionada a cadastros desatualizados.



O gráfico da feature month mostrou que o mês de maio (may) foi o que teve mais contratações. Os outros meses também tiveram contratações, exceto dezembro com uma baixa quantidade de adesão. Deve ser um comportamento normal (a abaixa adesão de dezembro) uma vez que, é o inicio das férias escolares e também das festas. Vale estudar se nesse período a abordagem aos clientes deve ser diferente dos demais meses ou mesmo, se o produto a ser ofertado não deve ser outro.



O gráfico da feature day_of_week não mostrou uma tendência na contratação do produto.

- Analisei também as features que possuem algum valor predominante na contratação dos produtos:

Calculando o percentual sobre o total de registros dos clientes que contratam o produto (y=yes)

```
print("Conteúdo das Features com mais registros com y=yes (total = ", dataset_sel.shape[0], ")")
perc = (dataset_sel_campaign.shape[0] / dataset_sel.shape[0]) * 100
print("campaign (1) com mais registros...................: ", dataset_sel_campaign.shape[0], " - % sobre total = ", perc)
perc = (dataset_sel_pdays.shape[0] / dataset_sel.shape[0]) * 100
print("pdays (999) com mais registros....................: ", dataset_sel_pdays.shape[0], " - % sobre total = ", perc)
perc = (dataset_sel_previous.shape[0] / dataset_sel.shape[0]) * 100
print("previous (0) com mais registros...................: ", dataset_sel_previous.shape[0], " - % sobre total = ", perc)
perc = (dataset_sel_job.shape[0] / dataset_sel.shape[0]) * 100
print("job (admin) com mais registros....................: ", dataset_sel_job.shape[0], " - % sobre total = ", perc)
perc = (dataset_sel_education.shape[0] / dataset_sel.shape[0]) * 100
print("edcucation (university.degree) com mais registros: ", dataset_sel_education.shape[0], " - % sobre total = ", perc)
perc = (dataset_sel_loan.shape[0] / dataset_sel.shape[0]) * 100
print("loan (no) com mais registros......................: ", dataset_sel_loan.shape[0], " - % sobre total = ", perc)
perc = (dataset_sel_contact.shape[0] / dataset_sel.shape[0]) * 100
print("contact (cellular) com mais registros.............: ", dataset_sel_contact.shape[0], " - % sobre total = ", perc)
```

```
Conteúdo das Features com mais registros com y=yes (total =  4639 )
campaign (1) com mais registros...................:  2299  - % sobre total =  49.5580944169002
pdays (999) com mais registros....................:  3672  - % sobre total =  79.15499029963354
previous (0) com mais registros...................:  3140  - % sobre total =  67.6870015089459
job (admin) com mais registros....................:  1388  - % sobre total =  29.920241431342966
edcucation (university.degree) com mais registros:  1824  - % sobre total =  39.31881871092908
loan (no) com mais registros......................:  3849  - % sobre total =  82.97046777322699
contact (cellular) com mais registros.............:  3852  - % sobre total =  83.03513688294892
```

Fiz a combinação das features com maiores percentuais:

Combinando as features com maior percentual sobre o total (y=yes)

```
print("Combinando as Features com mais registros com y=yes -> (total = ", dataset_sel.shape[0], ")")
dataset_sel_rept = dataset.query('y == "yes" and pdays == 999 and loan == "no" and contact == "cellular"')
perc = (dataset_sel_rept.shape[0] / dataset_sel.shape[0]) * 100
print("Total registros (pdays=999 / loan=no / contact=cellular): ", dataset_sel_rept.shape[0], " - % sobre total = ", perc)

dataset_sel_rept = dataset.query('y == "yes" and pdays == 999 and loan == "no"')
perc = (dataset_sel_rept.shape[0] / dataset_sel.shape[0]) * 100
print("Total registros (pdays=999 / loan=no /                ): ", dataset_sel_rept.shape[0], " - % sobre total = ", perc)

dataset_sel_rept = dataset.query('y == "yes" and pdays == 999 and contact == "cellular"')
perc = (dataset_sel_rept.shape[0] / dataset_sel.shape[0]) * 100
print("Total registros (pdays=999 /         / contact=cellular): ", dataset_sel_rept.shape[0], " - % sobre total = ", perc)

dataset_sel_rept = dataset.query('y == "yes" and loan == "no" and contact == "cellular"')
perc = (dataset_sel_rept.shape[0] / dataset_sel.shape[0]) * 100
print("Total registros (          / loan=no / contact=cellular): ", dataset_sel_rept.shape[0], " - % sobre total = ", perc)
```

```
Combinando as Features com mais registros com y=yes -> (total =  4639 )
Total registros (pdays=999 / loan=no / contact=cellular):  2459  - % sobre total =  53.00711360206941
Total registros (pdays=999 / loan=no /                ):  3060  - % sobre total =  65.96249191636129
Total registros (pdays=999 /         / contact=cellular):  2956  - % sobre total =  63.7206294460013
Total registros (          / loan=no / contact=cellular):  3189  - % sobre total =  68.74326363440396
```

As três features que apresentaram um valor dominante na contratação do produto foram:

- ✓ pdays
- ✓ loan
- ✓ contact

Para a feature pdays a maior parte dos clientes que contratou o produto estão com pdays=999, isso significa que são clientes que não foram contatados em outras campanhas. Isso pode significar que:

- Clientes não contatados (pdays=999) tendem a contratar o produto porque ainda não possuem nenhum produto de empréstimo bancário OU

- Clientes que foram contatados em outras campanhas (pdays <> 999) tendem a não contratar porque já possuem um produto bancários OU

- Clientes que foram contatados em outras campanhas (pdays <> 999) tendem a não contratar porque o produto não se adequa a sua necessidade ou não realmente não existe o interesse em contratar o produto

Para as duas opções, mencionadas acima, que não contrataram o produto vale realizar uma análise/estudo sob o motivo da recusa ao produto. Pode estar atrelado a forma como o cliente tem sido abordado ou as opções ofertadas (por exemplo, prazo, valor da parcela, juros, etc) ou o tipo de produto que está sendo ofertados (ex, empréstimo pessoal com ou sem garantia, consignado, consórcio, etc)

Para a feature loan a maior parte dos clientes que contratou o produto estão com feature=no, isso significa que são clientes que não possuem empréstimo. Isso pode significar que:

- Clientes que já possuem empréstimo tendem a não contratar outro produto de empréstimo OU

- O produto de empréstimo que está sendo ofertado não possui opções adequadas para clientes que já possuem empréstimo OU

- O tipo de produto ofertado não se adequa as necessidades dos clientes

Para a feature contact a maior parte dos clientes que contratou o produto estão com feature=cellular, isso pode significar que:

- O telefone é um meio que pode estar caindo em desuso OU

- O cadastro dos clientes com telefone pode estar desatualizado, que pode ocasionar o não contato com o cliente OU

- Por ser um telefone fixo, o horário de contato precisa ser diferenciado para encontrar as pessoas nas suas residências OU

Vale realizar uma análise/estudo nesse público com telefone fixo, para entender suas características.

Visualização Exploratória

Buscando identificar features mais relevantes ao target, utilizei o método de correlação de Pearson. O resultado mostrou que não existe nenhuma feature com uma forte correlação com a variável target (y).

```
In [212]: corr = features_encoded_transform.corr()
          corr.sort_values(["y"], ascending = False, inplace = True)
          print(corr.y)
```

```
y                              1.000000
poutcome_success               0.315641
previous                       0.228502
month_mar                      0.147819
contact_cellular               0.141648
month_sep                      0.129571
month_oct                      0.101290
job_student                    0.091489
job_retired                    0.088455
month_dec                      0.081423
month_apr                      0.080253
marital_single                 0.053514
education_university.degree    0.047089
job_admin.                     0.031474
poutcome_failure               0.028376
age                            0.027462
cons.conf.idx                  0.027215
job_unemployed                 0.013837
housing_yes                    0.012823
day_of_week_thu                0.012340
education_illiterate           0.007590
day_of_week_tue                0.007298
day_of_week_wed                0.007165
```

A mesma técnica foi aplicada para buscar a correlação entre todas as features (exceto a target).

```
features_corr.corr().style.format("{:.2}").background_gradient(cmap=plt.get_cmap('coolwarm'),axis=1)
```

	age	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	job_admin.	job_blue-collar
age	1.0	0.0067	-0.031	0.021	0.013	0.013	0.12	0.021	-0.0063	-0.089	-0.022
campaign	0.0067	1.0	0.051	-0.078	0.15	0.12	-0.0019	0.13	0.14	0.013	-0.0034
pdays	-0.031	0.051	1.0	-0.59	0.27	0.067	-0.075	0.3	0.37	-0.028	0.064
previous	0.021	-0.078	-0.59	1.0	-0.42	-0.2	-0.076	-0.46	-0.51	0.017	-0.053
emp.var.rate	0.013	0.15	0.27	-0.42	1.0	0.76	0.29	0.97	0.9	-0.02	0.049
cons.price.idx	0.013	0.12	0.067	-0.2	0.76	1.0	0.13	0.68	0.5	-0.037	0.069
cons.conf.idx	0.12	-0.0019	-0.075	-0.076	0.29	0.13	1.0	0.35	0.18	0.039	-0.094
euribor3m	0.021	0.13	0.3	-0.46	0.97	0.68	0.35	1.0	0.94	-0.02	0.04
nr.employed	-0.0063	0.14	0.37	-0.51	0.9	0.5	0.18	0.94	1.0	-0.021	0.057
job_admin.	-0.089	0.013	-0.028	0.017	-0.02	-0.037	0.039	-0.02	-0.021	1.0	-0.32
job_blue-collar	-0.022	-0.0034	0.064	-0.053	0.049	0.069	-0.094	0.04	0.057	-0.32	1.0
job_entrepreneur	0.032	-0.0027	0.018	-0.013	0.0076	0.0083	-0.031	0.018	0.023	-0.11	-0.1
job_housemaid	0.085	0.004	-0.0024	-0.012	0.038	0.029	0.037	0.037	0.029	-0.097	-0.088
job_management	0.064	-0.0088	0.0016	0.0057	-0.015	-0.024	-0.0022	4.7e-05	0.00059	-0.16	-0.15
job_retired	0.43	-0.0037	-0.071	0.064	-0.089	-0.037	0.067	-0.092	-0.12	-0.12	-0.11

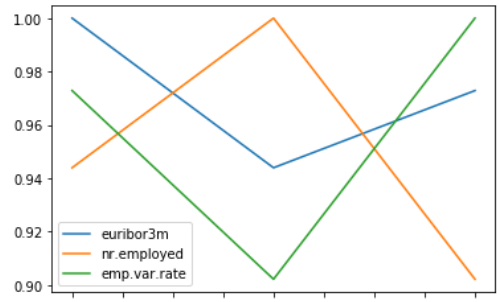
As features que apresentaram uma correlação mais forte (mais próxima de 1) foram plotadas num gráfico.

```
features_corr_maior = features_corr[['euribor3
features_corr_maior.corr().style.format("{:.2}
```

	euribor3m	nr.employed	emp.var.rate
euribor3m	1.0	0.94	0.97
nr.employed	0.94	1.0	0.9
emp.var.rate	0.97	0.9	1.0

```
features_corr_maior.corr().plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1482a4d1be0>
```



Analisando os gráficos constatei que as features, euribour3m e emp.var.rate possuem uma forte correlação.

Algoritmos e técnicas

<u>Técnicas e métodos que serão aplicados:</u>

Primeiramente foram utilizadas funções para visualização das features:

- ✓ filtros /seleções
- ✓ agrupamento de valores
- ✓ contagem dos domínios
- ✓ geração de gráficos

Execução de Funções para identificar e/ou tratar:

- ✓ valores ausentes
- ✓ registros duplicados
- ✓ média, mediana e desvio padrão
- ✓ normalização da escala das variáveis numéricas
- ✓ codificação das variáveis categóricas
- ✓ exclusão de features
- ✓ atribuição de valores aleatórios

Para variáveis categóricas foi adotado o método para substituir o valor ‘unknown' pelo domínio que mais se repetiu.

Para reduzir o tamanho do conjunto, buscando gerar um modelo mais performático foram utilizados os métodos:

- Métodos de correlação

- PCA (para reduzir dimensionalidade)

E por fim, serão treinados, testados e refinados modelos preditivos de aprendizagem supervisionada

Benchmark

Encontrei alguns trabalhos no GitHub, eles serão como modelo de referência para o modelo que estou propondo.

Modelo de Referencia 1:

Fonte: https://github.com/krishtanwani/bank-additional

Modelo: LogisticRegression

- ⇨ Acurácia: 0.896981467994
- ⇨ Tamanho do Teste: 30%

Modelo de Referencia 2 (essa fonte possui 3 modelos):

Fonte: https://github.com/juliencohensolal/BankMarketing

Modelo: SVC

- ⇨ Acurácia: 0.899193939983
- ⇨ Tamanho do teste: 25%

Modelo: RandomForestClassifier

- ⇨ Acurácia: 0.88798679227
- ⇨ Tamanho do teste: 25%

Modelo: LogisticRegression

- ⇨ Acurácia: 0.900320481694
- ⇨ Tamanho do teste: 25%

3 – Metodologia

Pré-Processamento

Para treinar e testar o modelo de machine learning foi necessário realizar um pré-processamento que envolveu:

- Exclusão dos registros duplicados

- Para todas as features categóricas, nos casos de valores ‘unknown’ foi atribuído o valor mais repetido

- Somente para a variável 'education' não foi possível adotar o método acima, pois, caso fosse adotado, todos os registros seriam marcados como ‘university.degree’ o que não faria sentido, pois, existem faixas de idades que não permitem. Para resolver esse problema foi realizado o seguinte processo:

- ✓ criação de uma feature nova para representar faixas de idades
- ✓ identificação, por faixa de idade, qual o domínio com a maior quantidade de registros
- ✓ para cada faixa, dos registros com valor ‘desconhecido’ foi atribuído o valor do domínio com maior quantidade
- ✓ - Transformar todas as variáveis categóricas em códigos para processamento do modelo

Como as variáveis numéricas possuíam valores com escalas diferentes, adotei o método para normalizar a escala dessas variáveis.

A feature duration foi excluída, pois, ela é obtida somente após a realização da campanha, ou seja, é um dado que nunca estará disponível no momento da execução do modelo preditivo

A feature ‘default’ também foi excluída, pois, ela possuía poucos registros com informações.

Para a feature housing foi adotado o método de atribuição aleatória para os registros com conteúdo ‘desconhecido’, pois, os dois valores possíveis possuem quantidade semelhantes.

Para identificar valores discrepantes (outliers) utilizei o método Turco para cada feature:

- ✓ cálculo do percentil 25 (Q1) e o percentil 75 (Q3)
- ✓ cálculo da amplitude interquantil x 1,5 (step)
- ✓ cálculo do menor valor (Q1 – step)
- ✓ cáclulo do maior valor (Q3 + step)

Feature	step = iqr * 1.5	Q1	Q1-step	Q3	Q3+step
Age	22,5	32	9,5	47	69,5
Campaign	3	1	-2	3	6
Pdays	0	999	999	999	999
Previous	0	0	0	0	0
emp.var.rate	4,8	-1,8	-6,6	1,4	6,2
cons.price.idx	1,3785	93,075	91,6965	93,994	95,3725
cons.conf.idx	9,45	-42,7	-52,15	-36,4	-26,95
euribor3m	5,4255	1,344	-4,0815	4,961	10,3865
nr.employed	193,5	5099,1	4905,6	5228,1	5421,6

São considerados outiliers os valores de cada feature que forem menor que Q1 – step ou maior que Q3 + step.

Nessa fase foi fundamental entender o comportamento das features, pois, caso contrário, as variáveis abaixo teriam registros considerados outliers indevidamente:

- age: valores < 32 ou > 69,5 seriam considerados outliers (468 linhas), não faz sentido, pois, são valores dentro da normalidade
- campaign: valores > 6 seriam considerados outliers (2406 linhas), não faz sentido, pois, podem existir clientes que foram contatados em mais de 6 campanhas
- pdays: valores < 999 ou > 999 seriam considerados outliers (1515 linhas), isso aconteceu porque existem muitos registros com valores 999 (cliente não contatados)
- previous: valores < 0 e > 0 seriam considerados outliers (5625 linhas), isso aconteceu porque a maioria dos clientes não foi contatada em outras campanhas (previous = 0)

As features emp.var.rate , cons.price.idx , euribor3m e nr.employed não apresentaram valores considerados outliers.

Somente a feature 'cons.conf.idx' apresentou valores que podem ser considerados outliers.

Para reduzir a dimensionalidade do conjunto, utilizei a correlação de Pearson e o método PCA (Principal Component Analysis.

Os modelos abaixo foram treinados e testados:

- Modelo Decision Tree Classifier

- Modelo Linear SGDClassifier

- Modelo Logistic Regression

- Modelo SVC

- Modelo Random Forest Classifier

E por fim, foram escolhidos para refinamento, os modelos que apresentaram a melhor acurácia:

- Modelo Linear SGDClassifier

- Modelo Logistic Regression

	Decision Tree Classifier	**Linear SGDClassifier**	**Logistic Regression**
Vantagens	- Fácil interpretar e converter para uma lógica se-então-senão, funciona para qualquer tipo de variável sem necessidade de conversão - baixo custo computacional	- Quando a quantidade de atributos é grande é melhor, pois, algoritmos mais complexos tendem a sofrer de sobreajuste	- Facilidade para lidar com variáveis independentes categóricas - Facilidade de classificação de indivíduos em categorias - Requer pequeno número de suposições
Desvantagens	- Casos raros (outliers) ou ruídos podem gerar resultados indesejados	- Existe um risco de relacionar estatisticamente coisas que na prática não tem relação nenhuma	- Ao contrário de regressão linear, regressão logística pode ser usada somente para prever funções discretas - Uma vez que o procedimento de estimação de parâmetros de regressão logística depende fortemente de ter um número suficiente de amostras para cada combinação de variáveis independentes, pequenas amostras podem levar a estimativas muito imprecisas dos parâmetros.
Usado em	- Sistemas de Recomendação	- Classificação de textos, filtragem de spam	- Em medicina, permite por exemplo determinar os fatores que caracterizam um grupo de indivíduos doentes em relação a indivíduos sãos - Em instituições financeiras, pode detectar os grupos de risco para a subscrição de um crédito

Implementação

1 - Eliminação dos registros duplicados

2 – Atribuição do valor para os casos ‘unknown’

3 – Tratamento da Feature ‘education’

3.1 – Criação das faixas de idade

```
df_age_education['faixa'] = df_age_education['education']
df_age_education.loc[df_age_education["age"] < 20, "faixa"] = "faixa de 10 a 19"
df_age_education.loc[(df_age_education["age"] >= 20) & (df_age_education["age"] < 30), "faixa"] = "faixa de 20 a 29"
df_age_education.loc[(df_age_education["age"] >= 30) & (df_age_education["age"] < 40), "faixa"] = "faixa de 30 a 39"
df_age_education.loc[(df_age_education["age"] >= 40) & (df_age_education["age"] < 50), "faixa"] = "faixa de 40 a 49"
df_age_education.loc[(df_age_education["age"] >= 50) & (df_age_education["age"] < 60), "faixa"] = "faixa de 50 a 59"
df_age_education.loc[(df_age_education["age"] >= 60) & (df_age_education["age"] < 70), "faixa"] = "faixa de 60 a 69"
df_age_education.loc[(df_age_education["age"] >= 70) & (df_age_education["age"] < 80), "faixa"] = "faixa de 70 a 79"
df_age_education.loc[(df_age_education["age"] >= 80) & (df_age_education["age"] < 90), "faixa"] = "faixa de 80 a 89"
df_age_education.loc[(df_age_education["age"] >= 90) & (df_age_education["age"] < 100), "faixa"] = "faixa de 90 a 99"
df_age_education.loc[df_age_education["age"] >= 100, "faixa"] = "faixa de 100 para cima"
```

3.2 – Visualização dos valores mais repetidos por faixa de idade (excluindo os ‘unknown)

3.3 – Atribuição dos valores por faixa:

```
dataset.loc[(dataset["age"] < 20) & (dataset["education"] == "unknown"), "education"] = maior_education[0]
dataset.loc[(dataset["age"] >= 20) & (dataset["age"] < 30) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[1]
dataset.loc[(dataset["age"] >= 30) & (dataset["age"] < 40) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[2]
dataset.loc[(dataset["age"] >= 40) & (dataset["age"] < 50) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[3]
dataset.loc[(dataset["age"] >= 50) & (dataset["age"] < 60) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[4]
dataset.loc[(dataset["age"] >= 60) & (dataset["age"] < 70) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[5]
dataset.loc[(dataset["age"] >= 70) & (dataset["age"] < 80) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[6]
dataset.loc[(dataset["age"] >= 80) & (dataset["age"] < 90) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[7]
dataset.loc[(dataset["age"] >= 90) & (dataset["age"] < 100) &
                    (dataset["education"] == "unknown"), "education"] =  maior_education[8]
```

4 – Atribuição aleatória para a feature ‘housing’:

5 –Método Turco para identificação e remoção de outliers

6 – Normalizando a escala das variáveis numérica (função MinMaxScaler)

7 – Codificando as variáveis categóricas para funcionamento do modelo preditivo (encoded).

8 – Cálculo da correlação pelo método Pearson.

9 – Cálculo do PCA.

10 – Implementação dos Modelos de Machine Learning:

10.1 – Divisão do Dataset em treino (80%) e teste (20%)

10.2 – Função para treino do Modelo

10.3 – Exemplo de Treino do Modelo:

```
learner = DecisionTreeClassifier()

sample_size = X_train.shape[0]
resultado = train_predict(learner, sample_size, X_train, y_train, X_test, y_test)
```

```
***** Treino *****
Acuracia.:  0.9945371961698993
Score....:  0.9744473155325869
***** Teste *****
Acuracia.:  0.8472870120304444
Score....:  0.33475935828877
```

Refinamento

Buscando obter melhores resultados, para os dois modelos que apresentaram a melhor acurácia, fiz alguns ajustes nos parâmetros dos modelos, obtendo uma melhora nos resultados.

Parâmetros passados para otimizar o modelo:

Parâmetro	SGDClassifier	LogisticRegression	LogisticRegression
random_state	15, 16, 17, 18, 19	1, 2, 3, 4	1, 2, 3, 4
Loss	hinge' 'log', 'modified_huber', 'squared_hinge', 'perceptron', 'squared_loss', 'huber', 'epsilon_insensitive', 'squared_epsilon_insensitive'	n/a	n/a

Penalty	none', 'l2', 'l1', 'elasticnet'	l1'	‘l2’
C	n/a	1, 2, 3, 4, 5, 6, 7, 8, 9	1, 2, 3, 4, 5, 6, 7, 8, 9
Solver	n/a	n/a	newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'

random_state (usada em todos os modelos): ‘semente’ geradora de números aleatórios usada pelo modelo. Para garantir que o modelo gere o mesmo resultado é necessário fixar um valor no random

loss (somente no modelo SGDClassifier): função de perda utilizada no modelo. Opções possíveis são: hinge', 'log', 'modified_huber', 'squared_hinge', 'perceptron', 'squared_loss', 'huber', 'epsilon_insensitive', 'squared_epsilon_insensitive'

penalty: penalidade a ser aplicada no modelo. No modelo SGDClassifier as opções são: none', 'l2', 'l1', 'elasticnet'. No modelo LogisticRegression as opções são: ‘l1’ e ‘l2’

C (somente no modelo LogisticRegression): Inversão da força de regularização; deve ser um flutuador positivo. Como nas máquinas de vetores de suporte, valores menores especificam uma regularização mais forte.

Solver (somente no modelo LogisticRegression): algoritmo. As opções são: ‘newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'

Melhores parâmetros retornados:

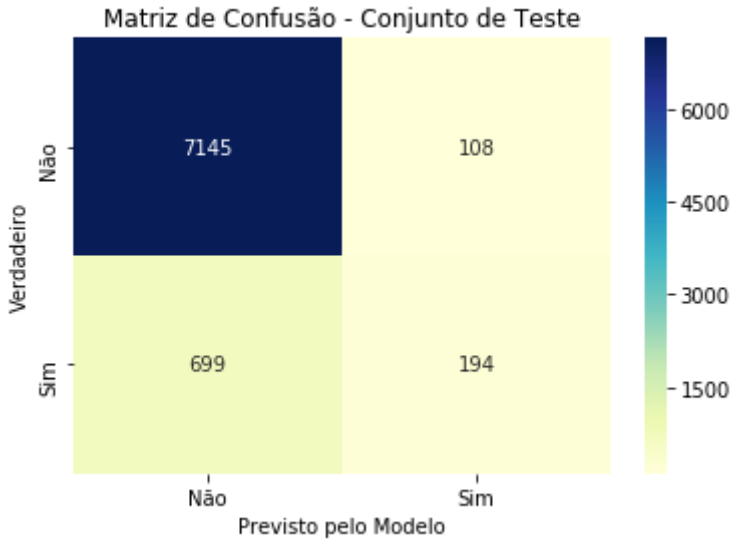
Parametro	**SGDClassifier**	**LogisticRegression**	**LogisticRegression**
random_state	16	1	3
Loss	log'	n/a	n/a
Penalty	l1'	l1'	l2'
C	n/a	3	2

Modelo melhor otimizado:

```
learner = LogisticRegression()

random_state = list(range(1, 5))
penalty = ['l2']
solver = ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
c = list(range(1, 10))

parameters = {'penalty':penalty,'solver':solver,'C':c, 'random_state': random_state}

scorer = 'accuracy'


train_predict_tuning(learner, parameters, scorer, X_train, y_train, X_test, y_test)
```

Resultados obtidos:

```
inicio:  15:01:29  - fim:  17:22:00

Modelo não Otimizado
------
Acuracia nos testes.......: 0.9008
F-score nos testes........: 0.3233
Precision-Score dos testes: 0.6412
Recall-Score dos testes...: 0.2161
Parametros................:  {'C': 1.0, 'class_weight': None, 'dual': False, 'fit_intercept': True, 'intercept_scaling':
1, 'max_iter': 100, 'multi_class': 'ovr', 'n_jobs': 1, 'penalty': 'l2', 'random_state': None, 'solver': 'liblinear', 'to
l': 0.0001, 'verbose': 0, 'warm_start': False}

Modelo Otimizado
------
Acuracia nos testes.......: 0.9009
F-score nos testes........: 0.3247
Precision-Score dos testes: 0.6424
Recall-Score dos testes...: 0.2172
Parametros................:  LogisticRegression(C=2, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=3, solver='sag', tol=0.0001,
          verbose=0, warm_start=False)
```



4 – Resultados

Modelo de avaliação e validação

O modelo escolhido foi o que apresentou melhor acurácia, ou seja, o modelo que obteve o maior percentual de acertos.

Para garantir que o modelo é confiável fiz uma quebra no conjunto real, deixando 80% para treino e 20% para teste. Os resultados mostraram que o modelo treinado está generalizando muito bem, pois, ele obteve ótimos resultados com os 20% de dados de teste.

Justificativa

Pelos modelos executados obtive melhores resultados (considerando a acurácia) que os modelos utilizados como benchmark:

Modelo	Logistic Regression	SVC	RandomForest Classifier	DecisionTree Classifier	SGDClassifier
Modelo de Referencia 1	0.8969	x	X	X	X
Modelo de Referencia 2	0.9003	0.8991	0.8879	X	X
Meu Projeto	0.9008 Otimizado: 0.9009	0.8997	0.8911	0.8465	0.8980 Otimizado: 0.9009

5 – Conclusão

A conclusão que eu cheguei foi que, para gerar um modelo de machine learning que consiga prever adequadamente um conjunto de dados é preciso:

1 – Primeiramente, entendimento claro do problema a ser resolvido

2 – Conhecimento do negócio relacionado ao problema. Esse conhecimento é fundamental para definir, localizar, avaliar e validar as melhores fontes de dados.

3– Conhecimento dos sistemas geradores das informações. É necessário entender como os sistemas armazenam os dados, a periodicidade que eles são gerados.

4 – Conhecimento de como funcionam os modelos e quais são os mais adequados para o problema a ser resolvido

5 – Conhecimento das métricas de avaliação dos modelos

6 – Conhecimento de linguagens de programação voltadas para machine learning (python, R, etc..)

7 – Conhecimento de métodos e técnicas tanto para análise, visualização e exploração de dados e principalmente para efetuar a tarefa de pré-processamento

O modelo executado obteve um maior numero de acerto na identificação dos clientes que não contrataram o produto, ou seja, ele pode ser bastante útil para não gastar/despender tempo nos clientes que não trarão retorno. Também obteve um bom rendimento na identificação dos clientes que realmente contratarão o produto. Para esses clientes inclusive vale analisar/estudar a possibilidade de oferta outros produtos e/ou produtos complementares (ex: seguro (de empréstimo, casa, carro, vida), cartão de crédito, consórcio, consignado, consultoria financeira).

Além do resultado do modelo executado, a análise exploratória dos dados identificou alguns comportamentos dos clientes que contrataram o produto que pode ser útil na definição da forma de abordagem ao cliente, nas ofertas e produtos a serem ofertados e principalmente no público a ser ofertado. Comportamentos identificados:

- Clientes que aderem a campanha não foram contatados em outras campanhas (pdays=999)

- Clientes que aderem a campanha não possuem empréstimo ativo (loan=no)

- Clientes que aderem a campanha tendem a utilizar cellular

Melhorias

Algumas melhorias podem ser adotadas nesse projeto:

- Aplicação de outras técnicas para detecção de outliers, redução de dimensionalidade do conjunto de dados e identificação de correlação entre variáveis

- Execução de outros modelos de machine learning

- Treino e testes com outros conjuntos de dados