PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data

Daniel Castanheira Martins

CLASSIFICAÇÃO E ESTUDO DAS ACEITAÇÕES DE ANÁLISES

Brasília
2021

Daniel Castanheira Martins

CLASSIFICAÇÃO E ESTUDO DAS ACEITAÇÕES DE ANÁLISES

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Brasília

2021

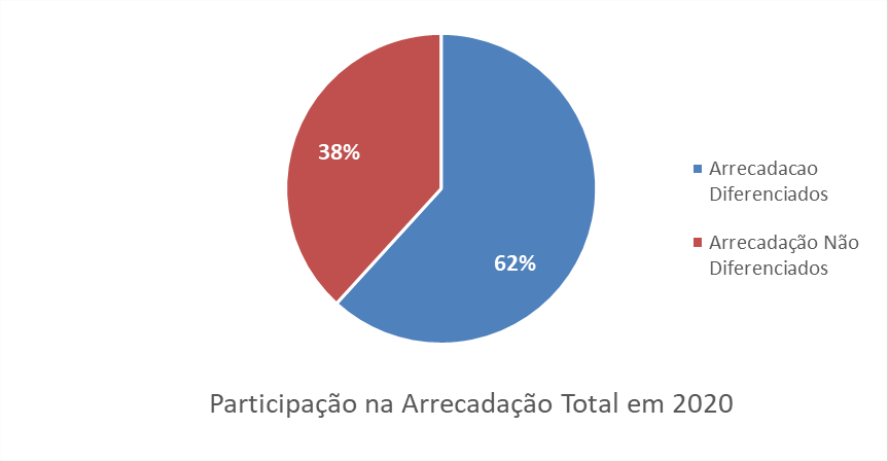
SUMÁRIO

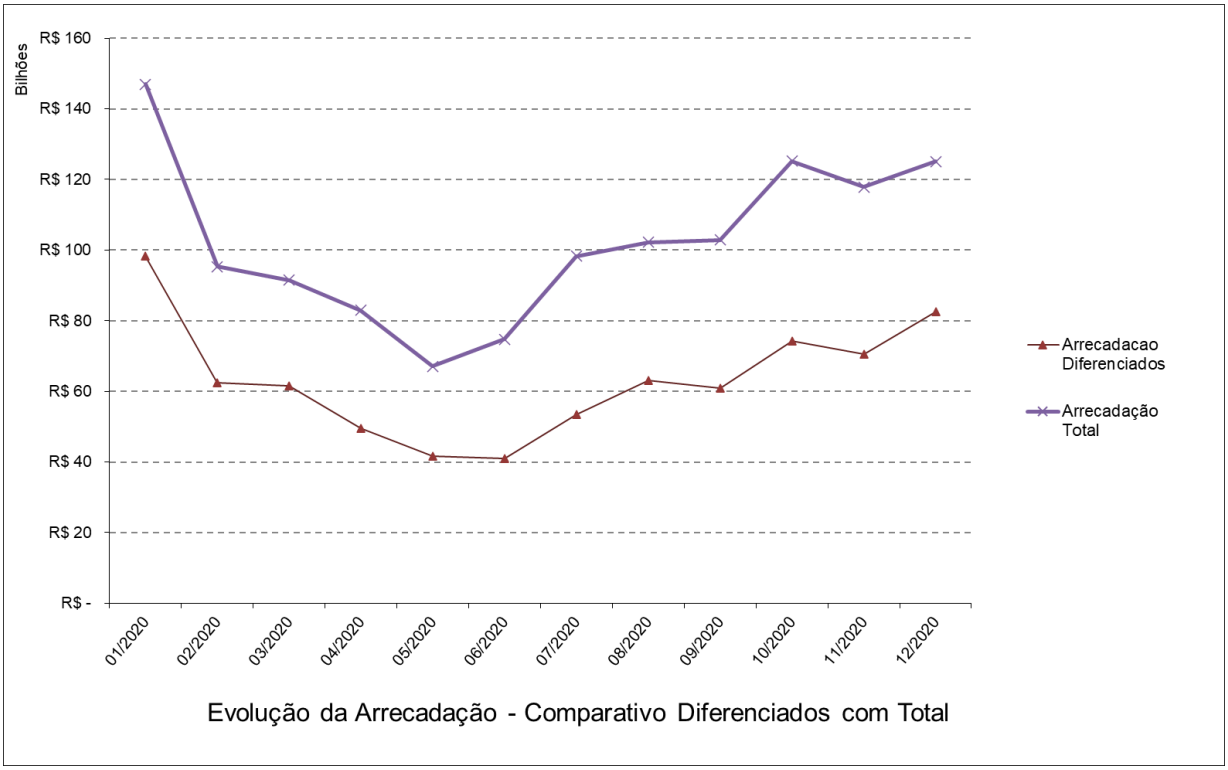
1. Introdução

1.1. Contextualização

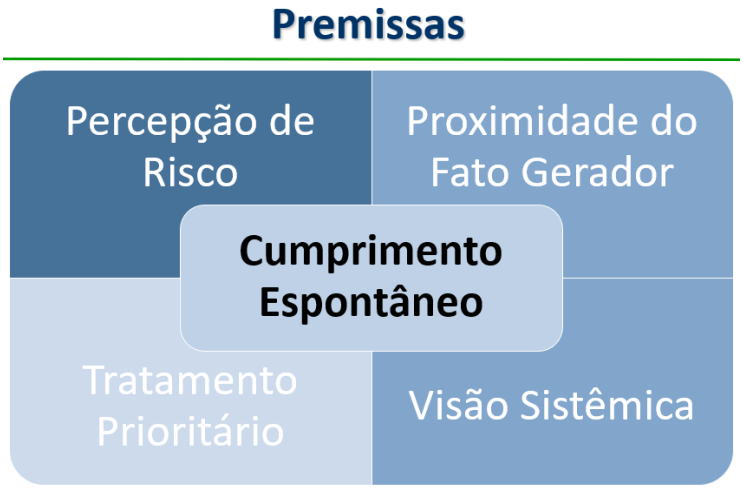
Na Receita Federal do Brasil, há uma coordenação voltada ao monitoramento dos maiores contribuintes do país. A arrecadação de tributos dessas empresas corresponde a aproximadamente 62% da total, e por esse motivo, existe a necessidade de se estabelecer uma atenção especial nesses casos.

Os gráficos abaixo possibilitam visualizar a importância e influência dessas pessoas jurídicas na arrecadação como um todo:





No quadro abaixo, demonstra-se as premissas pelas quais os maiores contribuintes são trabalhados:



Dentre essas premissas, como ponto focal de todas as outras, fica evidente o cumprimento espontâneo das obrigações tributárias como objetivo. Essa espontaneidade se caracteriza, na prática, na abertura de uma análise para estudo, e em uma comunicação às empresas quando confirmada alguma irregularidade, erro ou omissão fiscal, de forma que ele se regularize espontaneamente, sem necessidade de uma atuação fiscalizatória mais contundente, no primeiro momento.

Entretanto, em alguns casos, o contribuinte opta por não se autorregularizar. Nesses casos, as análises geradas pelo monitoramento dos maiores contribuintes são encaminhadas para outros processos de trabalho, que devem, idealmente, priorizar o tratamento dessas inconformidades.

1.2. O problema proposto

Devido a variados motivos, pode ocorrer a não aceitação das análises após o encaminhamento.

Esse trabalho consiste em um estudo das variáveis que podem melhorar ou não as aceitações, através de análises descritivas e modelagem dos dados obtidos, utilizando como base apurações de indicador de desempenho relacionado, no período abrangendo os anos de 2016 a 2020.

Visa-se, através desse projeto, a analisar alguns pontos com relação às aceitações ou não aceitações, como:

- Mostrar os níveis percentuais geograficamente, distribuídos nas Regiões Fiscais, como trabalhadas pela Receita Federal;
- Mostrar a evolução ao longo do tempo, de 2016 a 2020;
- Descobrir influências de atributos da base de dados, por exemplo:
    - Setores responsáveis pelo tratamento após os encaminhamentos;
    - Causa da distorção, consistindo na inconformidade tributária encontrada;
    - Tributo analisado na distorção.
- Obter, através de técnicas de Machine Learning, uma classificação que possa auxiliar na obtenção de melhores níveis no futuro;
- Selecionar o modelo com melhor avaliação, dentre as opções Árvore de Decisão, Random Forest, Adaboost e Nayve-Bayes.

2. Coleta de Dados

Os dados utilizados foram obtidos tendo como base as apurações, de 2016 a 2020, do indicador de desempenho K2 que, basicamente, consiste na razão entre os valores de arrecadação tributária esperados aceitos e os valores esperados encaminhados. Por razões de sigilo fiscal, essa base completa não será disponibilizada no repositório do trabalho. Porém, foi perfeitamente possível trabalhar sobre excertos das bases originais das apurações anuais, retirando-se dados sensíveis passíveis de possibilitar a identificação de contribuintes e valores envolvidos. Ainda houve inclusão de colunas de ano apurado, e outras obtidas através de cruzamento com banco de dados relacional da RFB proveniente de sistema transacional onde são registradas as análises. Essa junção será detalhada no capítulo referente a processamento e tratamento de dados.

2.1. Dataset Inicial

O dataset relacionado abaixo, resultante, como descrito acima, de um “slicing” dos dados de apurações do K2, foi utilizado para análises descritivas, especialmente para as temporais e geográficas:

Nome da coluna/campo	Descrição	Tipo
num_analise	Número da análise aberta após detectado erro, omissão ou irregularidade cometido pelo contribuinte.	String
ano_apuracao	Ano da apuração do K2, indicador de desempenho relacionado com as aceitações das análises.	String
RF	Região Fiscal que abriu a análise.	String
aceitacao	Aceitação ou não pelo outro processo de trabalho responsável pelo tratamento da distorção encontrada e para qual a análise foi encaminhada, quando não houve autorregularização nos contatos feitos pelo monitoramento dos maiores contribuintes.	String

2.2. Dataset para Modelagem

O próximo dataset foi obtido através de um cruzamento das bases de dados relacionais da Receita Federal, provenientes de sistema transacional utilizado para registrar análises de distorções, com o dataset anteriormente descrito, com fins de utilização nos modelos de Machine Learning:

Nome da coluna/campo	Descrição	Tipo
causa_distorcao	Consiste no erro, omissão ou irregularidade cometido pelo contribuinte, que serviu como causa para a abertura da análise.	String
responsavel_tratamento	Outro processo de trabalho responsável pelo tratamento da distorção encontrada e para qual a análise foi encaminhada, quando não houve autorregularização nos contatos feitos pelo monitoramento dos maiores contribuintes.	String
ca_nivel3	Código interno da RFB do tributo relacionado com a causa da distorção. Considerar a seguinte legenda para as abreviaturas: DV - Diversos; CD - Cadastro; RD - Receitas Diversas; RP - Receitas	String

	Previdenciárias.	
aceitacao	Aceitação ou não pelo outro processo de trabalho responsável pelo tratamento da distorção encontrada e para qual a análise foi encaminhada, quando não houve autorregularização nos contatos feitos pelo monitoramento dos maiores contribuintes.	String

Para melhor compreensão, cabe salientar que as Regiões Fiscais são compostas segundo o quadro a seguir:

a) SUPERINTENDÊNCIAS REGIONAIS DA RECEITA FEDERAL DO BRASIL E REGIÕES FISCAIS

Região Fiscal	Unidade	Sigla	Sede	UF	Jurisdição
1ª	Superintendência Regional da Receita Federal do Brasil da 1ª Região Fiscal	SRRF01	Brasília	DF	DF, GO, MT, MS e TO
2ª	Superintendência Regional da Receita Federal do Brasil da 2ª Região Fiscal	SRRF02	Belém	PA	PA, AP, RR, RO, AM e AC
3ª	Superintendência Regional da Receita Federal do Brasil da 3ª Região Fiscal	SRRF03	Fortaleza	CE	CE, PI e MA
4ª	Superintendência Regional da Receita Federal do Brasil da 4ª Região Fiscal	SRRF04	Recife	PE	PE, AL, RN e PB
5ª	Superintendência Regional da Receita Federal do Brasil da 5ª Região Fiscal	SRRF05	Salvador	BA	BA e SE
6ª	Superintendência Regional da Receita Federal do Brasil da 6ª Região Fiscal	SRRF06	Belo Horizonte	MG	MG
7ª	Superintendência Regional da Receita Federal do Brasil da 7ª Região Fiscal	SRRF07	Rio de Janeiro	RJ	RJ e ES
8ª	Superintendência Regional da Receita Federal do Brasil da 8ª Região Fiscal	SRRF08	São Paulo	SP	SP
9ª	Superintendência Regional da Receita Federal do Brasil da 9ª Região Fiscal	SRRF09	Curitiba	PR	PR e SC
10ª	Superintendência Regional da Receita Federal do Brasil da 10ª Região Fiscal	SRRF10	Porto Alegre	RS	RS

Link: https://receita.economia.gov.br/sobre/institucional/estrutura-organizacional/regimento-2020/arquivos-e-imagens/5-anexo-v-srrf.pdf

Quanto ao CA nível3, trata-se de abreviatura de código agregado nível 3. Existem 7 níveis de agregação para descrever os tributos internamente na Receita Federal. O CA nível 1 é o mais sintético e pouco explicativo, e o CE, código elementar, corresponde ao mais analítico. Para fins de utilização nos modelos, após várias ponderações e testes, o nível 3 se mostrou o mais adequado.

3. Processamento/Tratamento de Dados

O tratamento e processamento dos dados passou por várias etapas. Primeiramente, foi observado que as tabelas das apurações do indicador de desempenho K2, relacionados com as aceitações de análises, passaram por pequenas modificações ao longo dos anos, e isso impossibilitou uma união simples entre os dados dos diferentes anos.

3.1. Dataset Inicial

Com isso, a opção mais adequada mostrou-se ser a obtenção das colunas, provenientes das apurações, essenciais para análises exploratórias temporais e geográficas e para posterior cruzamento de dados (a fim de obter dataset para modelagem): número da análise (chave primária), região fiscal, aceitação e inclusão do atributo ano de apuração correspondente. Com isso, foi gerada uma tabela de 1509 registros, podendo ser encontrado como arquivo excel de nome “Aceitações_inicial.xlsx” no repositório do trabalho.

Foram detectados 23 números de análises duplicados, aparecendo em dois anos, nesse dataset. Optou-se por eliminar os registros mais antigos, considerando-se que os mais atualizados seriam os mais adequados e corretos, e pode-se consultar essas remoções no arquivo “registros_removidos.xlsx” no repositório. A eliminação dessas linhas foi feita manualmente, por serem poucos, ordenando os números de análises e pondo lado a lado com coluna ordenada já com os números duplicados removidos, permitindo assim a visualização dos duplicados e exclusão das linhas correspondentes, resultando em um dataset de 1486 linhas, com o nome de Aceitações_final.xlsx no repositório.

3.2. Dataset para Modelagem

Para os modelos de Machine Learning, foram escolhidos atributos passíveis de ter alguma influência nas aceitações ou não das análises encaminhadas a outros processos de trabalho: a causa da distorção, o responsável pelo tratamento e o CA nível 3, como detalhado no capítulo de coleta de dados.

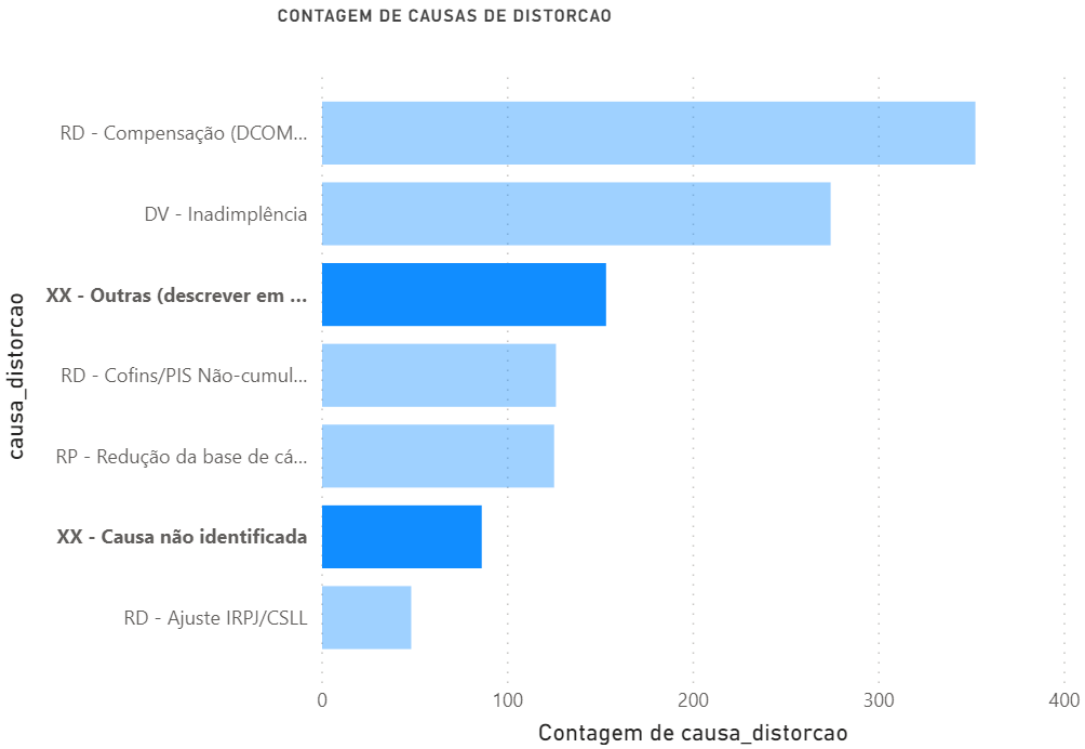
Para obter esses atributos, primeiramente, o dataset inicial, com a chave primária "número da análise", foi carregado no banco de dados relacional, possibilitando consulta por meio do código SQL abaixo (identificado no repositório como "SQL_sem_tratamento.sql":

```
select
   t2.nb_mcpj_anal_causa_distorcao as causa_distorcao,
   t3.nb_mcpj_anal_setor_resp as responsavel_tratamento,
   t5.cd_ca_nivel3 as ca_nivel3,
   t6.aceitacao as aceitacao
from mcpj.wf_mcpj_anals as t1
   join mcpj.wd_mcpj_anal_causa_distorcaos as t2
     on t1.nr_mcpj_anal_causa_distorcao = t2.nr_mcpj_anal_causa_distorcao
   join mcpj.wd_mcpj_anal_setor_resps as t3
     on t1.nr_mcpj_anal_setor_resp = t3.nr_mcpj_anal_setor_resp
   join dime.wd_rc_ca_nivel6 as t4
     on t1.nr_mcpj_csel_ca_n6_h_crit_sel = t4.nr_ca_nivel6
   join dime.wd_rc_ca_nivel3 as t5
     on t4.nr_ca_nivel3 = t5.nr_ca_nivel3
   join u01406263605.aceitacoes as t6
     on t1.dd_anal_num_analise = t6.num_analise
Where t1.dd_anal_num_analise in (select num_analise from
u01406263605.aceitacoes_v2)
```

A query resultou, como esperado, em uma tabela de 1486 linhas como no dataset inicial, chamada no repositório "Dataset sem tratamento.xlsx".

Entretanto, ao analisar alguns registros, foi possível verificar que alguns registros não seriam apropriados para modelagem, visto que não acrescentariam informação relevante para influenciar aceitações.

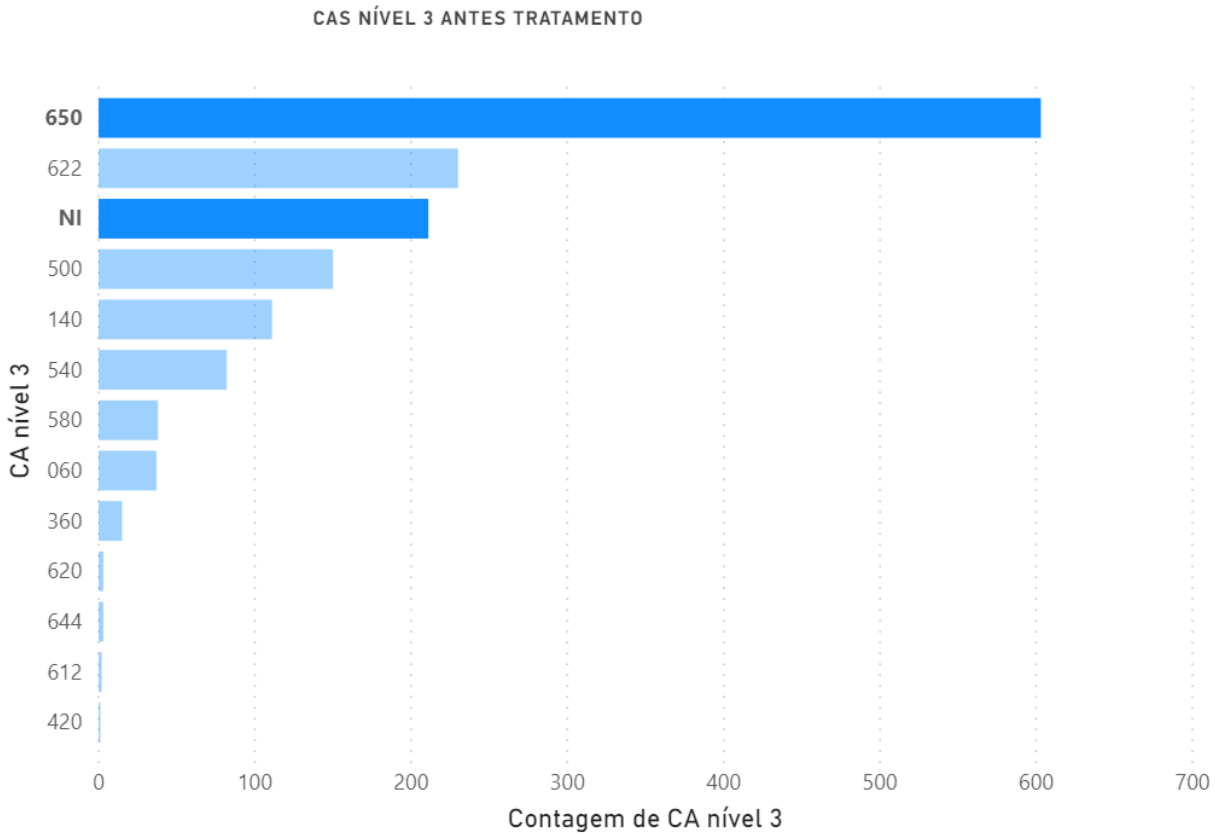
No atributo “Causa da Distorção”, foram identificados 153 registros “XX - Outras (descrever em observação)” e 86 “XX - Causa não identificada”, estando entre as sete causas mais frequentes, conforme gráfico abaixo:



Nesse caso, optou-se por remover todos esses registros, pois correspondem a um campo não informado. Seria desejável que a causa da distorção fosse identificada para uma classificação útil na prática.

No atributo “Responsável pelo Tratamento” havia 36 registros “Não informado”, também excluídos pelo mesmo motivo anteriormente citado.

No atributo “CA Nível 3”, o registro “NI” teve 211 ocorrências. Nesse caso, não se considerou interessante manter registros não informados, na mesma linha dos dois atributos anteriores. O código “650”, com 603 ocorrências, se trata de um CA nível 1, com o mais alto nível de agregação. Em algumas situações é possível informá-lo no sistema transacional e, por ser muito genérico, não caberia para uma conclusão na classificação. O gráfico abaixo ilustra a grande representação desses códigos de tributos no todo:

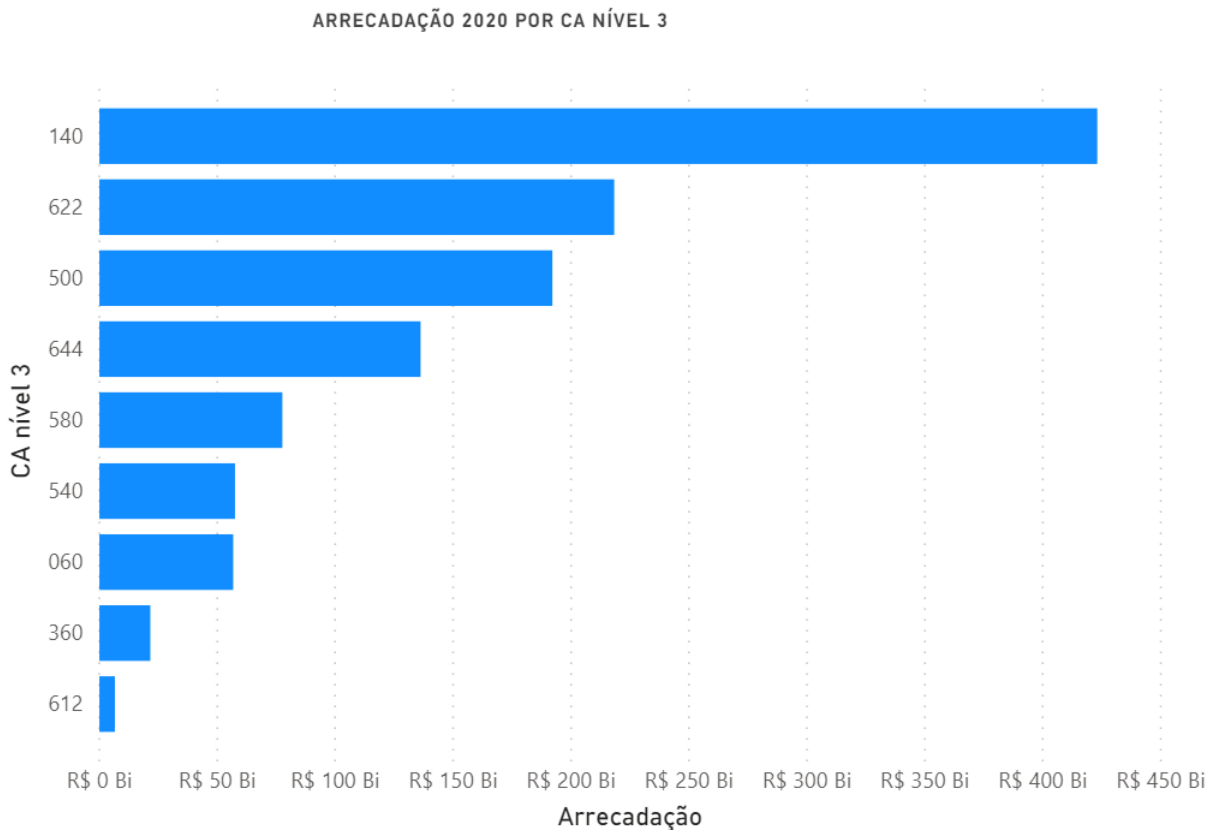


Essas duas situações receberam um tratamento especial, pois poderia ser feita uma inferência do tributo através do campo “Observação da Fase de Diagnóstico”. Não foi possível inclui-lo nos datasets do repositório, pois continha informações sensíveis, sujeitas a sigilo fiscal. Porém, foram examinados seus registros em busca de padrões e termos técnicos que pudessem auxiliar em uma inferência do CA nível 3, ou seja, a qual tributo estaria sendo tratada a análise da distorção identificada. Com isso, foram incluídas no SQL as seguintes condições:

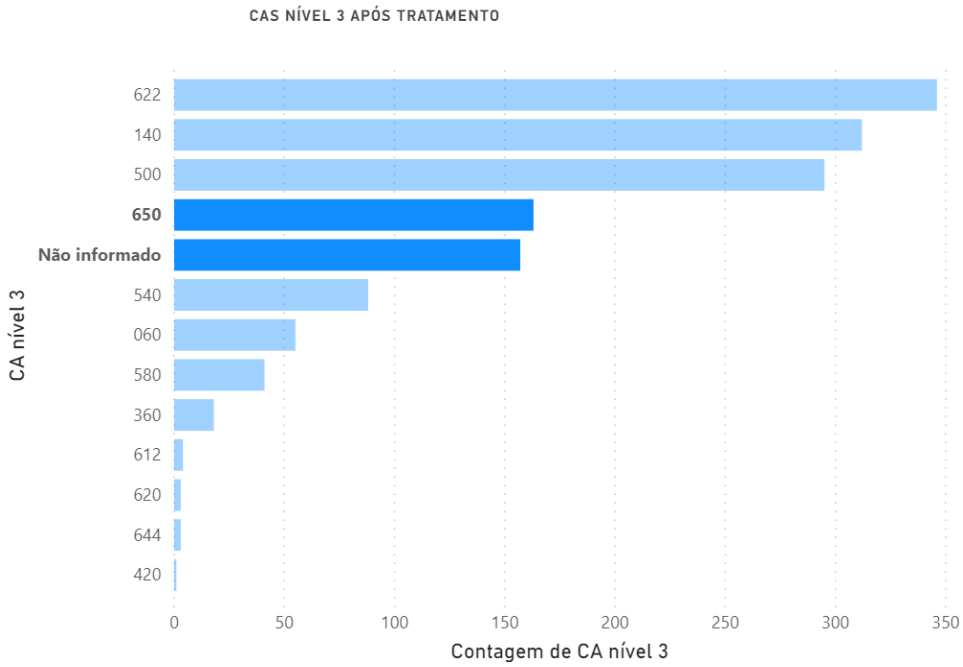
```
case
        when (t5.cd_ca_nivel3 != "NI" and t5.cd_ca_nivel3 != "650") then
t5.cd_ca_nivel3
        when (t7.nm_anal_obs_diagnostico ilike "%IRPJ%" or
t7.nm_anal_obs_diagnostico ilike "%ECF%" or t7.nm_anal_obs_diagnostico ilike
"%Lucro real%" or t7.nm_anal_obs_diagnostico ilike "%Lalur%" or
t7.nm_anal_obs_diagnostico ilike "%IRRF%" or t7.nm_anal_obs_diagnostico ilike
"%DIPJ%" or t7.nm_anal_obs_diagnostico ilike "%perdas não técnicas%" or
t7.nm_anal_obs_diagnostico ilike "%perdas não-técnicas%" or
```

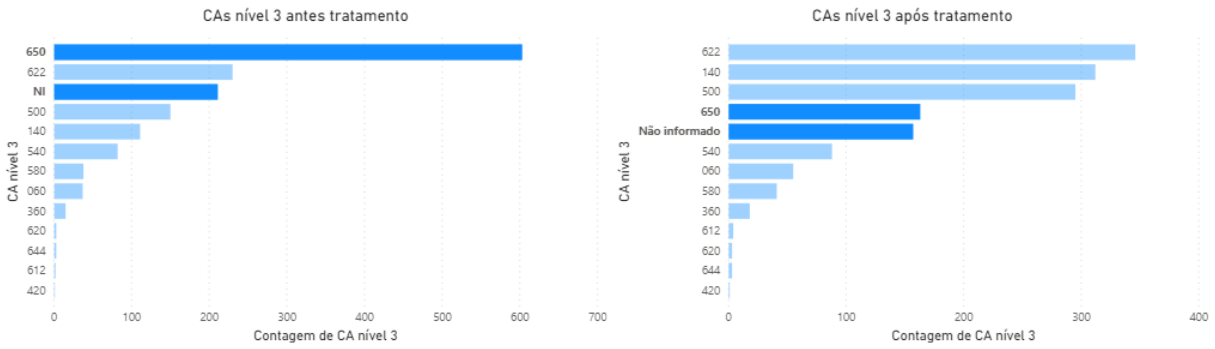
```
t2.nb_mcpj_anal_causa_distorcao ilike "%IRPJ%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%balancete%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%lucro%") then "140"
        when (t7.nm_anal_obs_diagnostico ilike "%previd%" or
t7.nm_anal_obs_diagnostico ilike "%GFIP%" or t7.nm_anal_obs_diagnostico ilike
"%GPS%"  or t7.nm_anal_obs_diagnostico ilike "%CPRB%" or
t7.nm_anal_obs_diagnostico ilike "%Dacon%" or t2.nb_mcpj_anal_causa_distorcao
ilike "%GPS%") then "622"
        when (t7.nm_anal_obs_diagnostico ilike "%Cofins%" or
t7.nm_anal_obs_diagnostico ilike "%EFD%" or t7.nm_anal_obs_diagnostico ilike
"%Sped Contribuições%" or t7.nm_anal_obs_diagnostico ilike "2172" or
t2.nb_mcpj_anal_causa_distorcao ilike "%Cofins%") then "500"
        when t7.nm_anal_obs_diagnostico ilike "%Pagamento Unificado%" then
"644"
        when t7.nm_anal_obs_diagnostico ilike "%CSLL%" then "580"
        when (t7.nm_anal_obs_diagnostico ilike "%Pis%" or
t7.nm_anal_obs_diagnostico ilike "%Pasep%" or t7.nm_anal_obs_diagnostico ilike
"%8109%") then "540"
        when (t7.nm_anal_obs_diagnostico ilike "%IPI%"  or
t2.nb_mcpj_anal_causa_distorcao ilike "%IPI%") then "060"
        when t7.nm_anal_obs_diagnostico ilike "%IOF%" then "360"
        when t7.nm_anal_obs_diagnostico ilike "%CIDE%" then "612"
        when t7.nm_anal_obs_diagnostico ilike "%Fundaf%" then  "620"
        when (t7.nm_anal_obs_diagnostico = "Não informado" or  t5.cd_ca_nivel3 =
"NI") then "Não informado"
        else "650" end as ca_nivel3,
```

A ordem dessas condições obedeceu a uma ordem de precedência, baseada no ranking da arrecadação de cada tributo no ano de 2020, conforme gráfico abaixo:



Após aplicadas essas condições, e sem aplicação de filtros, restaram 163 registros no CA 650 e 157 não informados, alcançando-se uma expressiva redução desses códigos. Além disso, outros alcançaram maior representatividade no todo. Nos gráficos abaixo, mostra-se essa evolução na qualidade dos dados.





Feita essa mitigação dos impactos desses registros, optou-se por excluir as linhas onde foram encontrados, para gerar informações de tributos significativas.

A versão final do código em SQL aplicou todos os tratamentos explicados, com o cálculo, já demonstrado, do CA nível 3, e filtros, excluindo os dados não desejados. Essa query está no repositório com o nome “SQL_com_tratamento.sql” e apresentada abaixo, sendo destacadas em negrito as operações realizadas (no filtro, Outros corresponderia ao CA 650):

```
with base as
(select
    t2.nb_mcpj_anal_causa_distorcao as causa_distorcao,
    t3.nb_mcpj_anal_setor_resp as responsavel_tratamento,
    case
    when (t5.cd_ca_nivel3 != "NI" and t5.cd_ca_nivel3 != "650") then t5.cd_ca_nivel3
    when (t7.nm_anal_obs_diagnostico ilike "%IRPJ%" or t7.nm_anal_obs_diagnostico ilike "%ECF%" or t7.nm_anal_obs_diagnostico ilike "%Lucro real%" or t7.nm_anal_obs_diagnostico ilike "%Lalur%" or t7.nm_anal_obs_diagnostico ilike "%IRRF%" or t7.nm_anal_obs_diagnostico ilike "%DIPJ%" or t7.nm_anal_obs_diagnostico ilike "%perdas não técnicas%" or t7.nm_anal_obs_diagnostico ilike "%perdas não-técnicas%" or t2.nb_mcpj_anal_causa_distorcao ilike "%IRPJ%" or t2.nb_mcpj_anal_causa_distorcao ilike "%balancete%" or t2.nb_mcpj_anal_causa_distorcao ilike "%lucro%") then "140"
    when (t7.nm_anal_obs_diagnostico ilike "%previd%" or
```

```
t7.nm_anal_obs_diagnostico ilike "%GFIP%" or t7.nm_anal_obs_diagnostico
ilike "%GPS%" or t7.nm_anal_obs_diagnostico ilike "%CPRB%" or
t7.nm_anal_obs_diagnostico ilike "%Dacon%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%GPS%") then "622"
  when (t7.nm_anal_obs_diagnostico ilike "%Cofins%" or
t7.nm_anal_obs_diagnostico ilike "%EFD%" or t7.nm_anal_obs_diagnostico
ilike "%Sped Contribuições%" or t7.nm_anal_obs_diagnostico ilike "2172" or
t2.nb_mcpj_anal_causa_distorcao ilike "%Cofins%") then "500"
  when t7.nm_anal_obs_diagnostico ilike "%Pagamento Unificado%" then
"644"
  when t7.nm_anal_obs_diagnostico ilike "%CSLL%" then "580"
  when (t7.nm_anal_obs_diagnostico ilike "%Pis%" or
t7.nm_anal_obs_diagnostico ilike "%Pasep%" or t7.nm_anal_obs_diagnostico
ilike "%8109%") then "540"
  when (t7.nm_anal_obs_diagnostico ilike "%IPI%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%IPI%") then "060"
  when t7.nm_anal_obs_diagnostico ilike "%IOF%" then "360"
  when t7.nm_anal_obs_diagnostico ilike "%CIDE%" then "612"
  when t7.nm_anal_obs_diagnostico ilike "%Fundaf%" then  "620"
  when (t7.nm_anal_obs_diagnostico = "Não informado" or  t5.cd_ca_nivel3 =
"NI") then "Não informado"
  else "Outros" end as ca_nivel3,
      t6.aceitacao as aceitacao
from mcpj.wf_mcpj_anals as t1
  join mcpj.wd_mcpj_anal_causa_distorcaos as t2
    on t1.nr_mcpj_anal_causa_distorcao = t2.nr_mcpj_anal_causa_distorcao
  join mcpj.wd_mcpj_anal_setor_resps as t3
    on t1.nr_mcpj_anal_setor_resp = t3.nr_mcpj_anal_setor_resp
  join dime.wd_rc_ca_nivel6 as t4
    on t1.nr_mcpj_csel_ca_n6_h_crit_sel = t4.nr_ca_nivel6
  join dime.wd_rc_ca_nivel3 as t5
    on t4.nr_ca_nivel3 = t5.nr_ca_nivel3
  join u01406263605.aceitacoes_v2 as t6
```

```
    on t1.dd_anal_num_analise = t6.num_analise
  join mcpj.wd_mcpj_anals as t7
    on (t1.dd_anal_num_analise = t7.dd_anal_num_analise and t1.nr_mcpj_anal = t7.nr_mcpj_anal)
where t1.dd_anal_num_analise in (select num_analise from u01406263605.aceitacoes_v2) )

select
causa_distorcao,
responsavel_tratamento,
ca_nivel3,
aceitacao
from base
where causa_distorcao not in ("XX - Causa não identificada", "XX - Outras (descrever em observação)")
and responsavel_tratamento not in ("Não informado")
and ca_nivel3 not in ("Não informado", "Outros")
```

O dataset de 1486 linhas antes do tratamento passou a ter, após essas exclusões, 1009 linhas, com o nome de “dataset_para_classificacao.xlsx” no repositório, estando preparado, dessa forma, para utilização nos modelos de Machine Learning.

4. Análise e Exploração dos Dados

Para a plotagem dos gráficos, foi dada preferência à utilização do Power BI, sendo possível consultar o relatório em arquivo disponibilizado no repositório com o nome “Visualização dos Dados de Aceitações.pbix”, que pode ser acessado e submetido a interações, se for de interesse do leitor, através da ferramenta.

4.1. Dataset inicial

O dataset, sendo composto somente de atributos categóricos, resultou no seguinte resultado após aplicado o comando describe() do pandas. O número da análise foi excluído da descrição, por ser apenas um número identificador.

1.1 - Describe simples

```
aceitacoes.iloc[:,1:].describe()
```

	ano_apuracao	rf	aceitacao
count	1486	1486	1486
unique	5	10	2
top	2016	RF09	Não
freq	539	323	745

Adicionalmente, realizaram-se os seguintes describes:

1.2 - Describe por Ano de Apuração

```
aceitacoes[['ano_apuracao', 'aceitacao']].groupby(['ano_apuracao'], as_index=False).count().describe()
```

	aceitacao
count	5.000000
mean	297.200000
std	151.283509
min	148.000000
25%	194.000000
50%	295.000000
75%	310.000000
max	539.000000

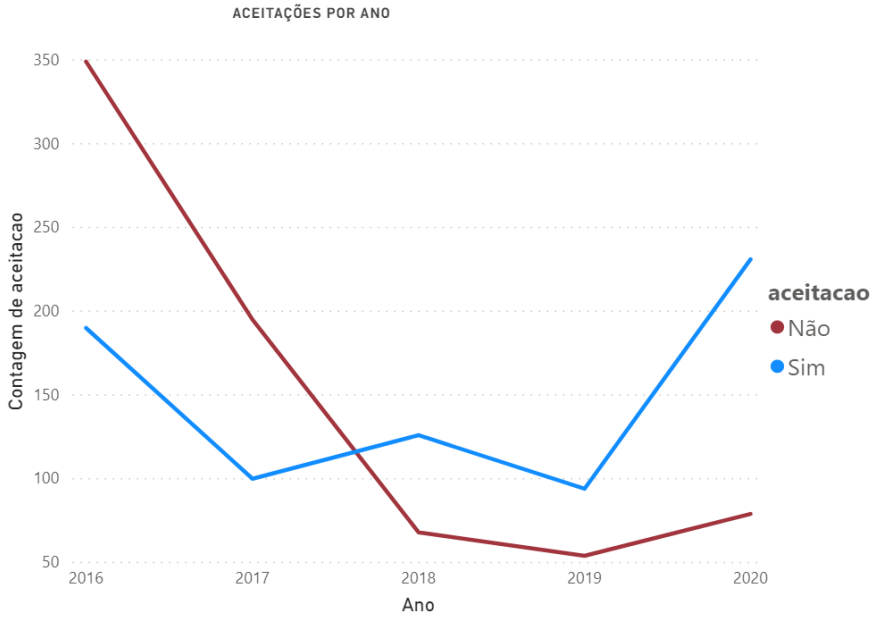
1.3 - Describe por Regiões Fiscais

```
aceitacoes[['rf', 'aceitacao']].groupby(['rf'], as_index=False).count().describe()
```

	aceitacao
count	10.000000
mean	148.600000
std	88.589189
min	58.000000
25%	97.250000
50%	112.500000
75%	162.250000
max	323.000000

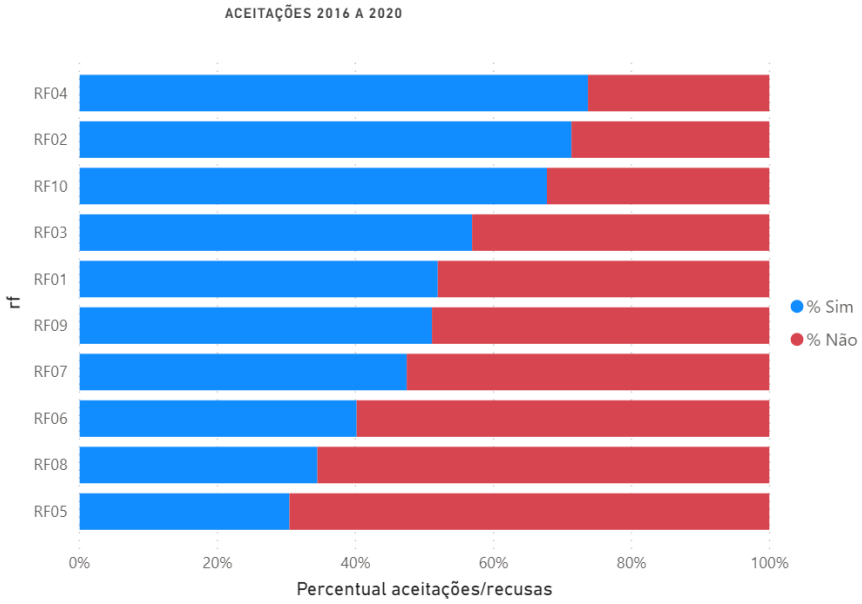
Os códigos acima podem ser consultados no notebook do Jupyter disponível no repositório com o nome de “Describes.ipynb”.

No gráfico temporal a seguir, constata-se que houve aumento expressivo em 2020 de aceitações, em detrimento de recusas.

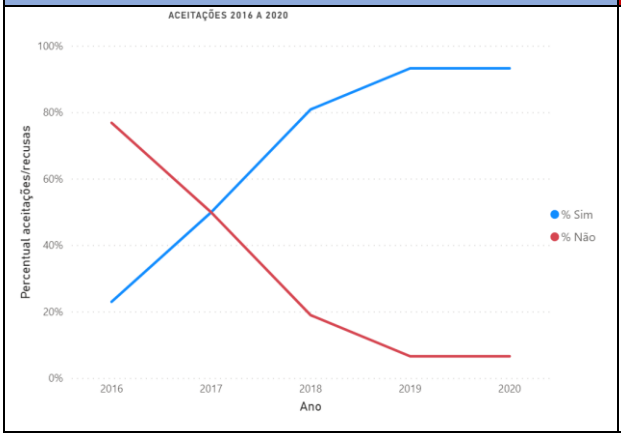
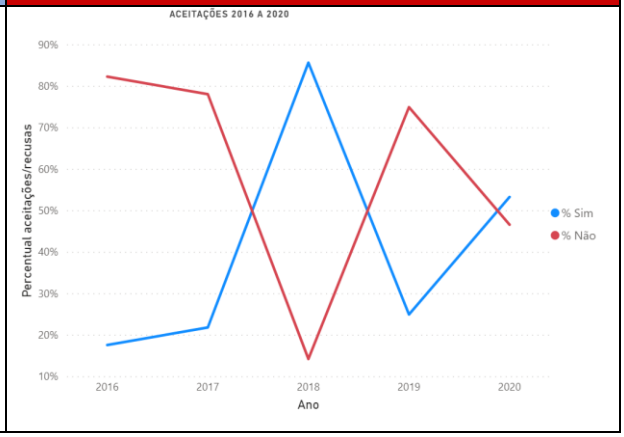


Para comparação das Regiões Fiscais e seu desempenho em obter anuências aos encaminhamentos, foram selecionados, a princípio, todos os anos escopo do projeto, ou seja, 2016 a 2020. Alguns testes demonstraram que a escolha de apenas um dos períodos poderia, em alguns casos, trazer poucas ocorrências para uma ou outra região.

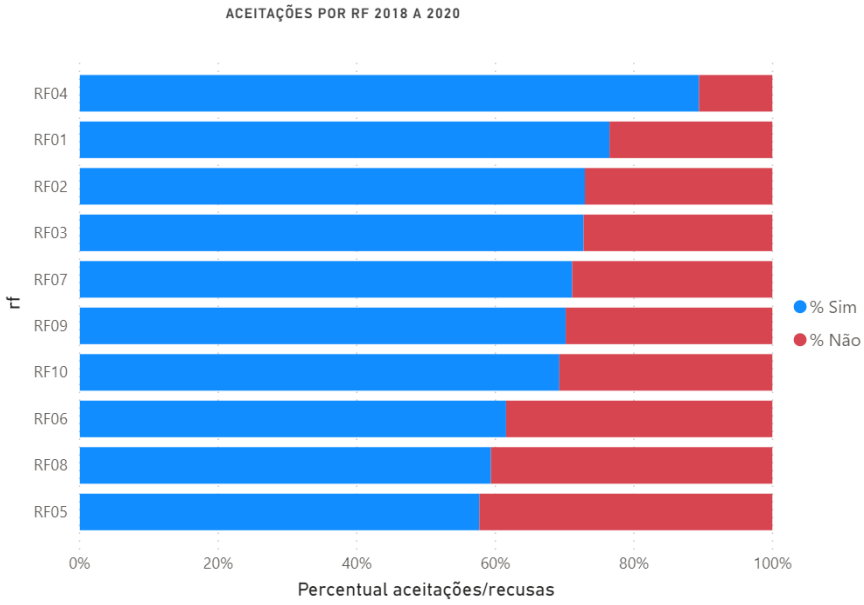
Através da interação com o Power BI, é possível fazer um drill down no gráfico, selecionando uma das RFs, por exemplo, e vendo como foi a evolução ao longo dos anos. Como essa seria uma análise temporal, após esse drill, pode ser selecionado o tipo de gráfico de linhas para melhor se ajustar a esse objetivo.



Exemplos de drill down para as RFs 04 e 05:

RF04	RF05
	

Para obter uma visão mais atualizada, um gráfico com os anos de 2018 a 2020 também foi gerado:



Observa-se pelos dois gráficos de Aceitações por Regiões Fiscais que as Regiões Fiscais 04, 02 e 03 se destacaram com percentuais maiores de aceitações ao longo dos anos, enquanto as 06, 08 e 05, os menores.

4.2. Dataset para Modelagem

Aplicando describes ao dataset, foram retornados:

2.1 - Describe simples

```
dataset.describe()
```

	causa_distorcao	responsavel_tratamento	ca_nivel3	aceitacao
count	1009	1009	1009	1009
unique	38	11	11	2
top	RD - Compensação (DCOMP ou Previdenciária)	X-ort	622	Não
freq	270	221	305	528

Além desse, foram aplicados outros describes abaixo:

2.2 - Describe por Causa da Distorção

```
dataset[['causa_distorcao', 'aceitacao']].groupby(['causa_distorcao'], as_index= False).count().describe()
```

	aceitacao
count	38.000000
mean	26.552632
std	55.494678
min	1.000000
25%	2.000000
50%	4.500000
75%	17.750000
max	270.000000

2.3 - Describe por Responsável pelo Tratamento

```
dataset[['responsavel_tratamento', 'aceitacao']].groupby(['responsavel_tratamento'], as_index= False).count().describe()
```

	aceitacao
count	11.000000
mean	91.727273
std	80.772633
min	1.000000
25%	6.000000
50%	109.000000
75%	142.000000
max	221.000000

2.4 - Describe por CA Nível 3

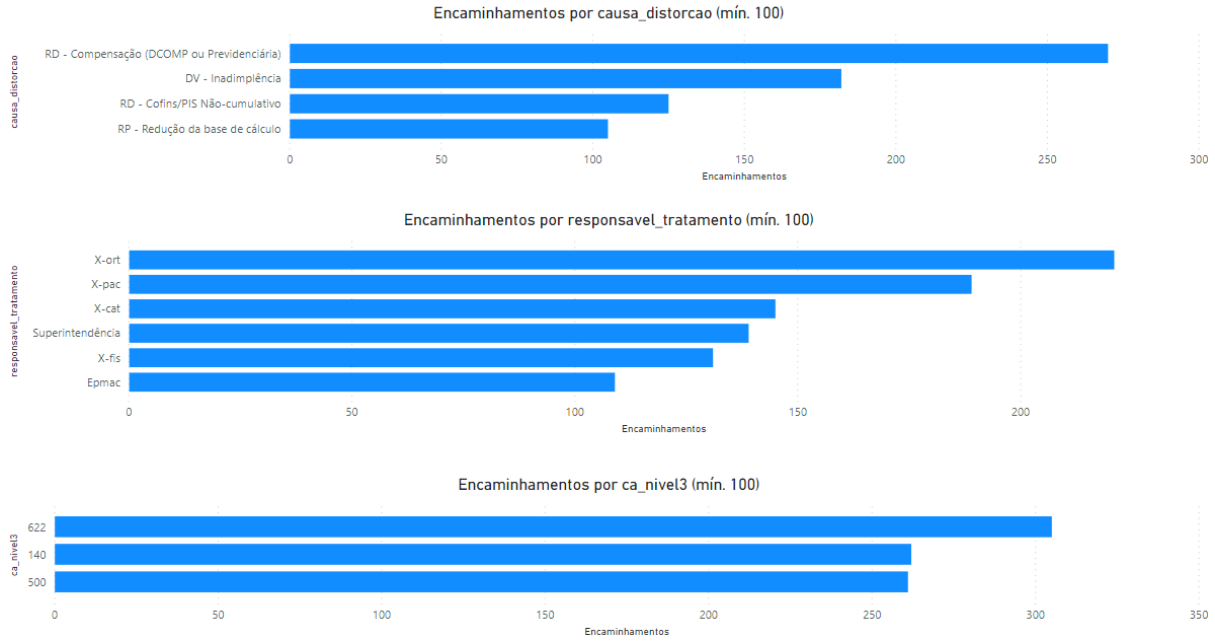
```
dataset[['ca_nivel3', 'aceitacao']].groupby(['ca_nivel3'], as_index= False).count().describe()
```

	aceitacao
count	11.000000
mean	91.727273
std	120.836328
min	1.000000
25%	3.000000
50%	37.000000
75%	164.500000
max	305.000000

Os códigos acima podem ser consultados no notebook do Jupyter disponível no repositório com o nome de “Describes.ipynb”.

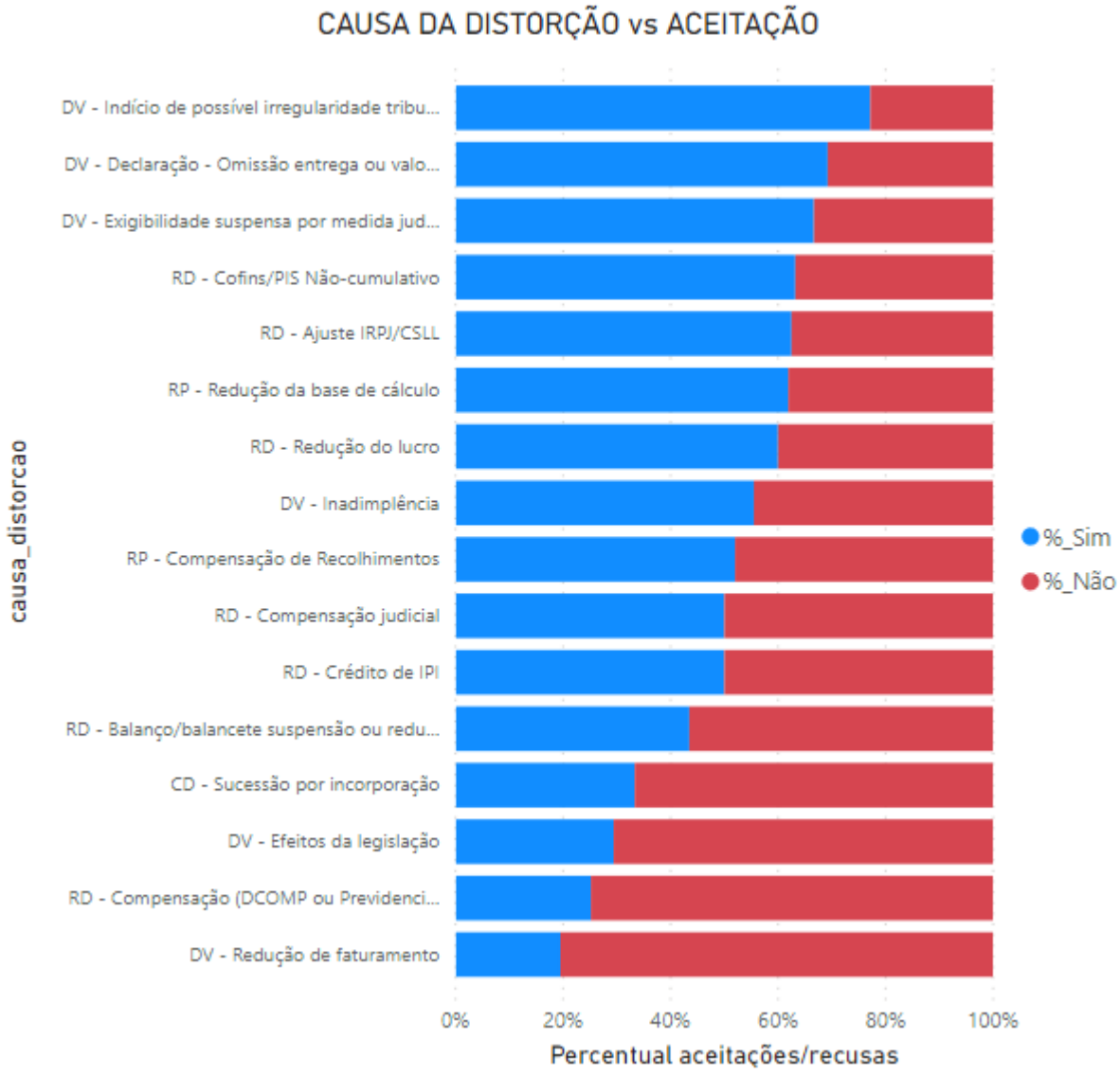
Complementando, ranquear a causa da distorção, responsável pelo tratamento e CA nível 3 pela contagem pode também contribuir para deduções

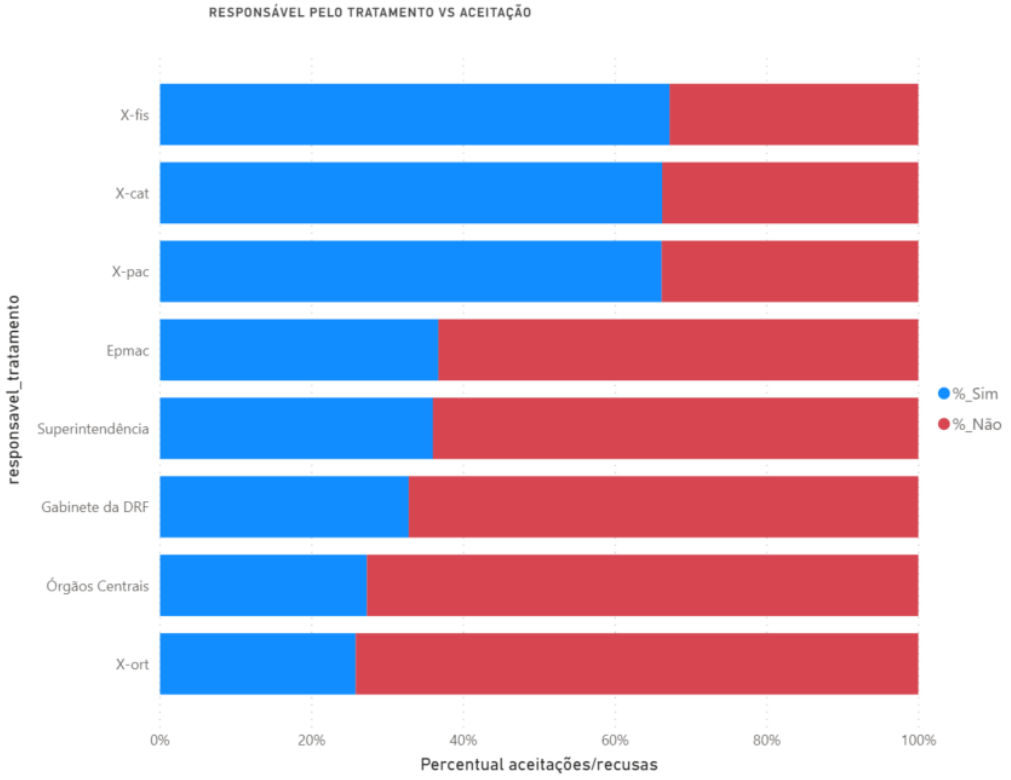
juntamente com outras análises. Para os gráficos abaixo, foram filtrados registros com um número mínimo de 100 eventos dentre os 1009 encaminhamentos:

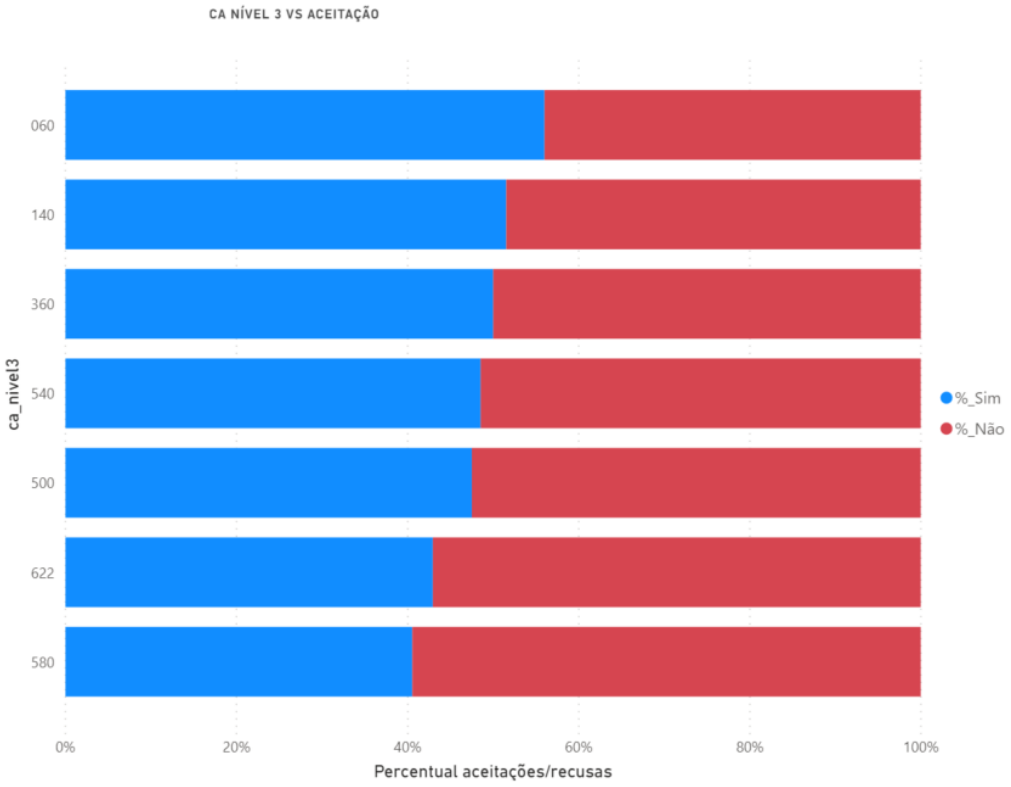


Para as visualizações das aceitações, primeiramente, foi definido um número de ocorrência mínimo de registros de 9 para que os dados do dataset fossem considerados nos gráficos, visando a diminuir o número de barras, desconsiderando dados irrelevantes. Esse número não foi escolhido apenas por esse motivo. Na fase de modelagem, o número mínimo de amostras por folha (min_samples_leaf) também foi o mesmo: o que apresentou melhor acurácia em combinação com os argumentos definidos para outros parâmetros.

Para cada atributo, foi elaborado um gráfico para visualizar como cada um gerou aceitações ou recusas:


CAUSA DA DISTORÇÃO vs ACEITAÇÃO
causa_distorcao
DV - Indício de possível irregularidade tribu...
DV - Declaração - Omissão entrega ou valo...
DV - Exigibilidade suspensa por medida jud...
RD - Cofins/PIS Não-cumulativo
RD - Ajuste IRPJ/CSLL
RP - Redução da base de cálculo
RD - Redução do lucro
DV - Inadimplência
RP - Compensação de Recolhimentos
RD - Compensação judicial
RD - Crédito de IPI
RD - Balanço/balancete suspensão ou redu...
CD - Sucessão por incorporação
DV - Efeitos da legislação
RD - Compensação (DCOMP ou Previdenci...
DV - Redução de faturamento
%_Sim
%_Não
0%
20%
40%
60%
80%
100%
Percentual aceitações/recusas

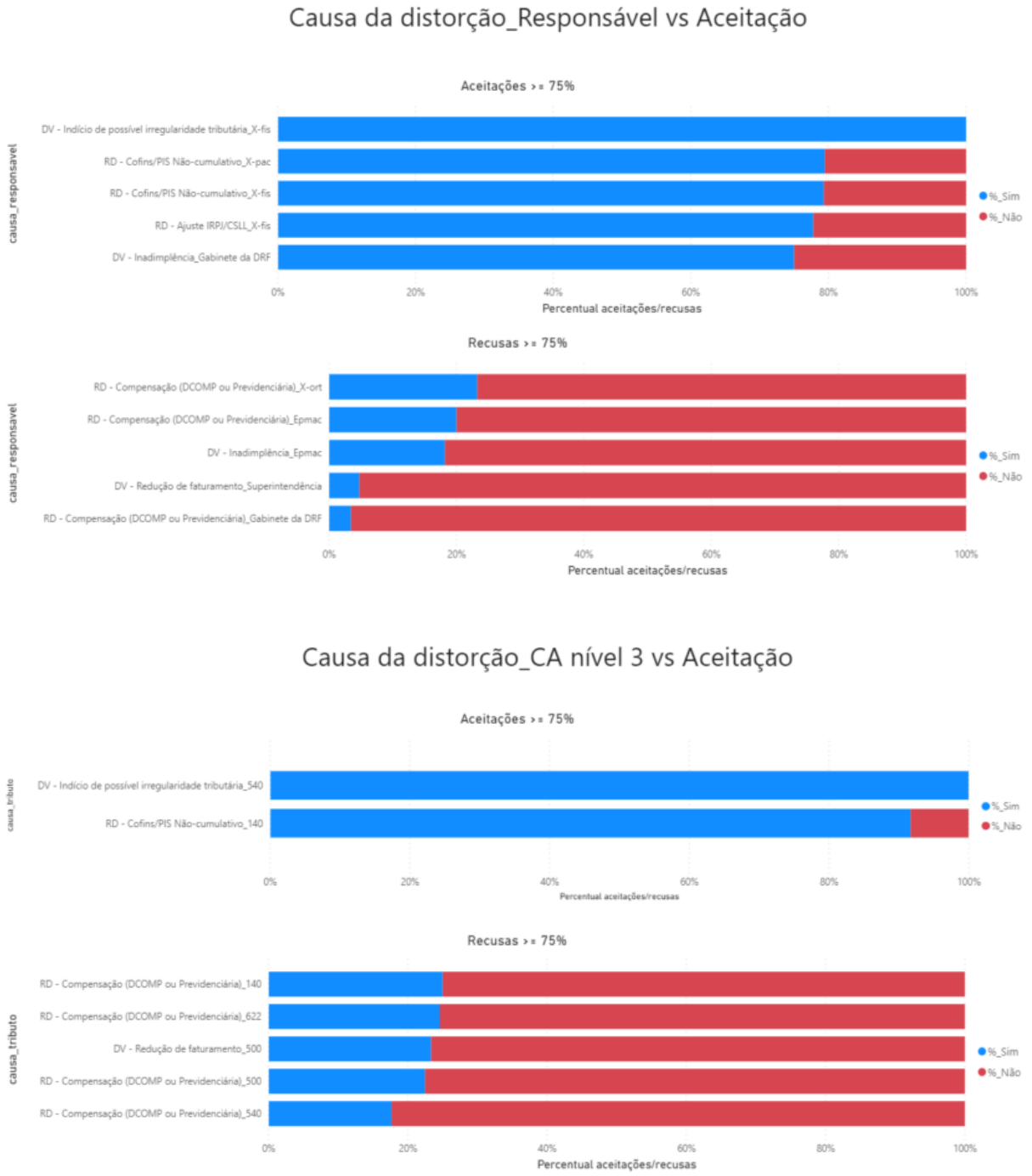


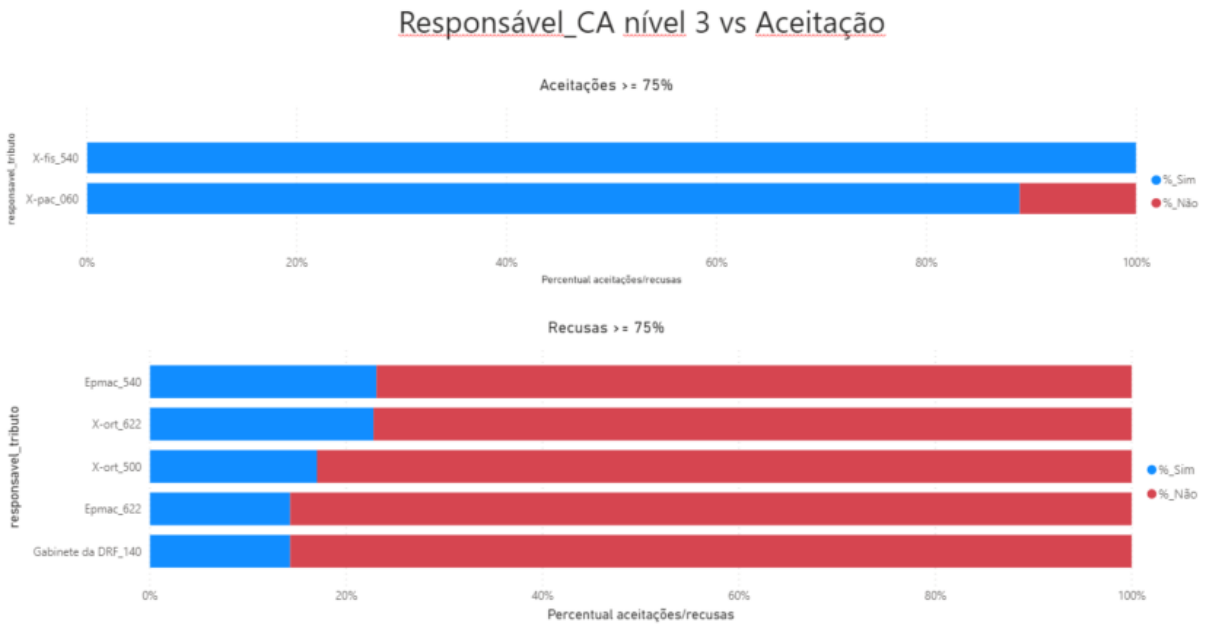


A quantidade reduzida de atributos possibilitou uma combinação de dois dos três, para cada um dos gráficos abaixo, concatenando-se os registros em novas

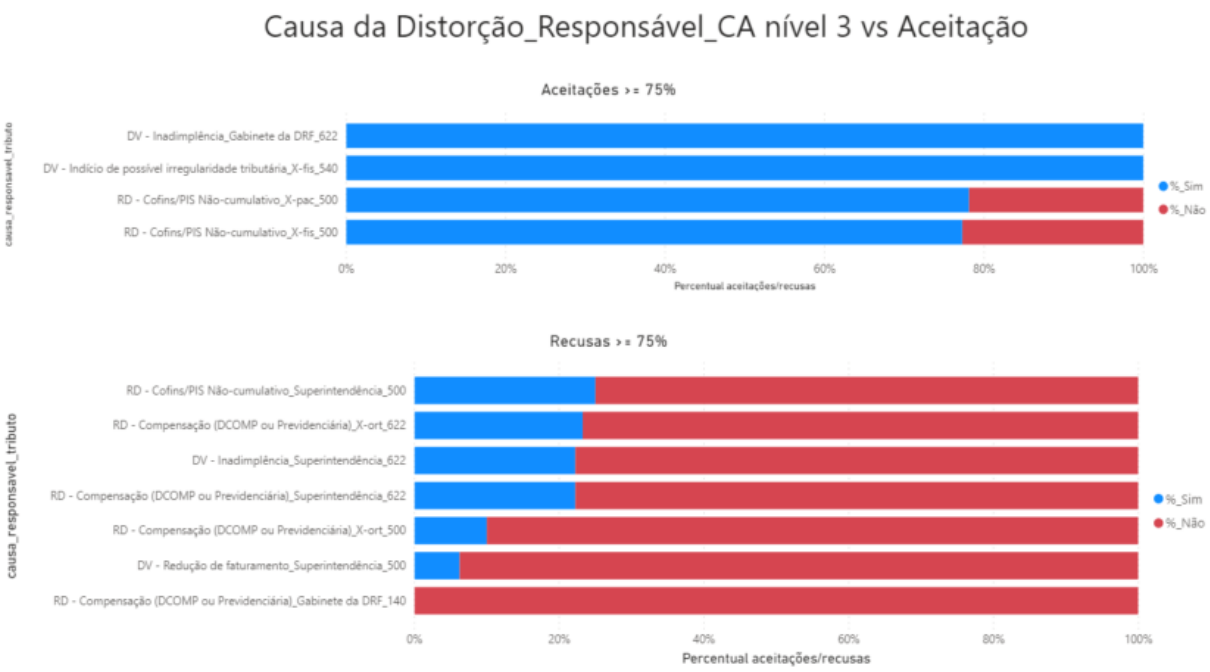
colunas criadas no Power BI, a fim de obter informação que pode ser útil para a proposta desse trabalho.

Esses gráficos mostram as combinações com percentuais maiores ou iguais a 75% nas aceitações ou recusas, para que não houvesse um excesso de barras ou se causasse poluição visual, trazendo apenas as mais importantes.





Na mesma linha e utilizando-se os mesmos critérios dos gráficos acima, entretanto usando a combinação de todos os atributos, criou-se o seguinte gráfico:



Após análise dos gráficos e dos describes, pode-se avaliar positivamente ou negativamente, conforme abaixo:

Melhor aceitação	Pior aceitação
• a causa da distorção “DV - Indício de possível irregularidade tributária” foi bem aceita, tendo seus resultados mais expressivos ocorrendo quando encaminhada para a X-fis com o CA 540. Combinado com o CA 060, teve destaque positivo;	• a causa da distorção “RD - Compensação (DCOMP ou Previdenciária)” é a mais frequente nos encaminhamentos a outros processos de trabalho, e não apresenta bons índices de aceitação em várias combinações, o que leva a refletir se as equipes têm alguma dificuldade ou obstáculo em tratar essa modalidade;
• a causa da distorção “RD - Cofins/PIS Não-cumulativo” teve bons índices, quando combinada com o atributo de responsável X-pac ou X-fis e CA 500. Combinado com o CA 140, obteve bons resultados.	• a causa da distorção “DV – Redução do Faturamento” também se destacou nos gráficos como pouco aceita, principalmente quando os encaminhamentos são feitos à Superintendência com o CA 500. Quando analisado com os CAs, o 540 também se destacou negativamente;
• a causa “DV – Inadimplência”, quando encaminhada a gabinete de DRF com o CA 622, tem bom nível de aceitações. Em outras combinações, tem desempenho mediano;	• O CA 622 é o mais frequente, entretanto, é o segundo mais recusado. Obtém mais aceitações na combinação do item logo à direita no quadro;
• X-fis, X-pac e X-cat são os responsáveis pelo tratamento que mais aceitam.	• A X-ort é a que mais recusa, o que tem um impacto significativo, por ser a destinatária mais frequente de encaminhamentos.

5. Criação de Modelos de Machine Learning

Os modelos de Machine Learning podem ser consultados no Jupyter Notebook “ML_Aceitação.ipynb” disponível no repositório. Nele, utilizaram-se e avaliaram-se os algoritmos de classificação “Decision Tree Classifier”, “Random Forest”, “Adaboost” e “GaussianNB” (neste último, NB abrevia Nayve Bayes), tendo como base o Dataset para Modelagem, explicado ao longo desse trabalho.

Buscaram-se classificações tendo como classe alvo binária a Aceitação, podendo assumir os valores Sim ou Não, com o objetivo de auxiliar a tomada de decisão quando do encaminhamento a outros processos de trabalho através da escolha dos melhores caminhos, condições e combinações para aceitação.

O dataset foi carregado, vetorizado e particionado em 75% para a base de treinamento e 25% para a base de testes, para aplicação nos cálculos de avaliação dos modelos. Também se avaliou a acurácia por meio de cross validation com 4 partições.

5.1. Decision Tree Classifier

Os argumentos e os parâmetros que geraram melhores avaliações para esse modelo foram:

- random_state = 42
- criterion = 'gini'
- max_depth = 7
- min_samples_leaf = 9

Foram realizados vários testes e análises da visualização da árvore de decisão para definir esses valores, até se alcançarem os melhores índices na avaliação. Também houve uma tentativa de se utilizar o parâmetro min_samples_split, que não produziu alterações na pontuação, exceto quando maior ou igual a 24, entretanto, piorando os scores. Por fim, alterações no max_depth ou no min_samples_leaf também não melhoraram as notas, ou seja não existe uma faixa de valores produzindo melhores resultados, sendo necessário usar esses valores específicos.

Avaliação do modelo:

```
Acurácia (base de treinamento): 0.6878306878306878
============================================================
       Acurácia de Previsão e Classification Report

Acurácia de previsão: 0.6996047430830039
              precision    recall  f1-score   support

         Sim       0.75      0.61      0.67       127
         Não       0.67      0.79      0.72       126

    accuracy                           0.70       253
   macro avg       0.71      0.70      0.70       253
weighted avg       0.71      0.70      0.70       253

============================================================
      Matriz de Confusão

     Sim (prev)  Não (prev)
Sim          77          50
Não          26         100
```

Cross validation:

```
Acurácia por Cross Validation

0.6547619047619048
```

5.2. Random Forest

Argumentos e parâmetros:

- random_state = 42
- criterion = 'gini'
- max_depth = 7
- min_samples_leaf = 9
- n_estimators = 17
- n_jobs = -1

Alguns argumentos foram os mesmos para manter uma comparabilidade, e ainda assim, obtiveram melhores pontuações nos testes da Random Forest. O valor específico de 17 para os n_estimators (número de árvores combinadas para a classificação final) produziram os melhores resultados, não havendo melhora se alterado.

Avaliação do modelo:

```
Acurácia (base de treinamento): 0.667989417989418
============================================================
       Acurácia de Previsão e Classification Report

Acurácia de previsão: 0.6837944664031621
              precision    recall  f1-score   support

         Sim       0.68      0.71      0.69       127
         Não       0.69      0.66      0.67       126

    accuracy                           0.68       253
   macro avg       0.68      0.68      0.68       253
weighted avg       0.68      0.68      0.68       253

============================================================
      Matriz de Confusão

     Sim (prev)  Não (prev)
Sim          90          37
Não          43          83
```

Cross validation:

```
Acurácia por Cross Validation

0.6349206349206349
```

5.3. Adaboost

Argumentos e parâmetros, tomando por estimador base uma árvore de decisão igual à usada no item 5.1:

- base_estimator = DecisionTreeClassifier(random_state=42, criterion='gini', max_depth = 7, min_samples_leaf = 9)
- random_state = 42
- n_estimators = 10
- algorithm='SAMME'

O valor específico de 10 para n_estimators foi o que produziu melhores notas.

Avaliação do modelo:

```
Acurácia (base de treinamento): 0.7341269841269841
============================================================
       Acurácia de Previsão e Classification Report

Acurácia de previsão: 0.6877470355731226
              precision    recall  f1-score   support

         Sim       0.68      0.72      0.70       127
         Não       0.70      0.65      0.67       126

    accuracy                           0.69       253
   macro avg       0.69      0.69      0.69       253
weighted avg       0.69      0.69      0.69       253

============================================================
      Matriz de Confusão

     Sim (prev)  Não (prev)
Sim          92          35
Não          44          82
```

Cross validation:

```
Acurácia por Cross Validation

0.6402116402116402
```

5.4. Nayve-Bayes

Não foi necessário definir argumentos de parâmetros para essa classificação, que foi mantida no projeto mesmo após apresentar pontuações baixas, apenas como um exemplo de uso do algoritmo.

Avaliação do modelo:

```
Acurácia (base de treinamento): 0.5912698412698413
============================================================
       Acurácia de Previsão e Classification Report

Acurácia de previsão: 0.541501976284585
              precision    recall  f1-score   support

         Sim       0.53      0.88      0.66       127
         Não       0.62      0.20      0.30       126

    accuracy                           0.54       253
   macro avg       0.58      0.54      0.48       253
weighted avg       0.58      0.54      0.48       253

============================================================
      Matriz de Confusão

     Sim (prev)  Não (prev)
Sim         112          15
Não         101          25
```

Cross validation:

```
Acurácia por Cross Validation

0.5621693121693121
```

Como demonstrado acima, o melhor classificador para o dataset proposto foi o “Decision Tree Classifier” e o pior, o “Gaussian Nayve-Bayes”. O “Adaboost”

também foi considerado um bom classificador, com scores bem próximos dos alcançados pelo Classificador de Árvore de Decisão. Todos, exceto o “Nayve-Bayes, alcançaram notas próximas de 70%, sendo passíveis de serem utilizados.

Para fins práticos, utilizaremos, na apresentação de resultados, a árvore de decisão gerada pelo “Decision Tree Classifier”, como veremos a seguir. A única ressalva desse classificador, mais bem avaliado no geral, seria uma nota menor de 61% na revocação (número de true positives / (número de true positives + número de false negatives). Portanto, dentre os classificados como Sim na realidade, esse modelo, em comparação com os outros, apresentou um número maior de falsos negativos.

6. Apresentação dos Resultados

O fluxo de trabalho desse projeto pode ser descrito através do Canvas, proposto por Jasmine Vasandani, abaixo:

Data Science Workflow Canvas*

Title: Classificação e Estudo das Aceitações de Análises

1 Problem Statement What problem are you trying to solve? What larger issues do the problem address? **Definição do Problema** Necessidade de melhorar os índices de aceitações de análises, quando há encaminhamento a outros processos de trabalho.	**2 Outcomes/Predictions** What prediction(s) are you trying to make? Identify applicable predictor (X) and/or target (y) variables. **Resultados e Previsões** Atributos preditivos: causa da distorção, responsável pelo tratamento e CA nível 3. Atributos alvo: Aceitação podendo ser Sim (1) ou Não (0). Visa-se a uma classificação e estudo das variáveis que possam levar a aceitações ou recusas de análises encaminhadas a outros processos de trabalho.	**3 Data Acquisition** Where are you sourcing your data from? Is there enough data? Can you work with it? **Coleta de Dados** Tabelas com apurações do indicador K2, relacionado com os encaminhamentos e aceitações. Banco de dados relacional da RFB. Os datasets apresentam pouco mais de 1000 registros, atendendo a mínimo exigido para realização do trabalho.
4 Modeling What models are appropriate to use given your outcomes? **Modelagem** Utilização dos modelos de Machine Learning aplicáveis a classificações: ● Decision Tree Classifier; ● Random Forest; ● Adaboost; ● Gaussian Nayve-Bayes.	**5 Model Evaluation** How can you evaluate your model's performance? **Avaliação do Modelo** Para avaliar os modelos, utilização de particionamento do dataset, seguido de medição das acurácias da base de treinamento e teste, classification report para de teste, bem como cross validation.	**6 Data Preparation** What do you need to do to your data in order to run your model and achieve your outcomes? **Preparação dos Dados** Necessidade de eliminação de registros não informados e similares; Inferência do tributo escopo da análise, através de campo com descrição do diagnóstico realizado, e eliminação posterior dos excessivamente genéricos e não informados.

Conceptualized by Jasmine Vasandani using notes from General Assembly's Data Science Immersive. Format inspired by Business Model Canvas.

6.1. Resultados destacados da Análise Exploratória

Abaixo são apresentadas algumas conclusões após análise exploratória.

Pontos fortes e fracos:

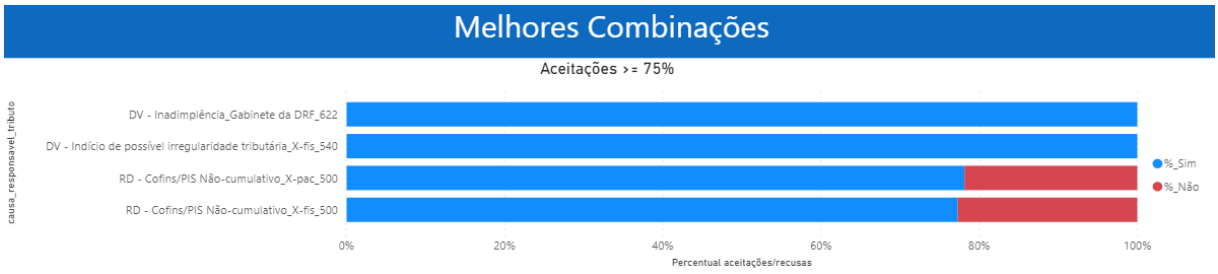
Pontos Fortes

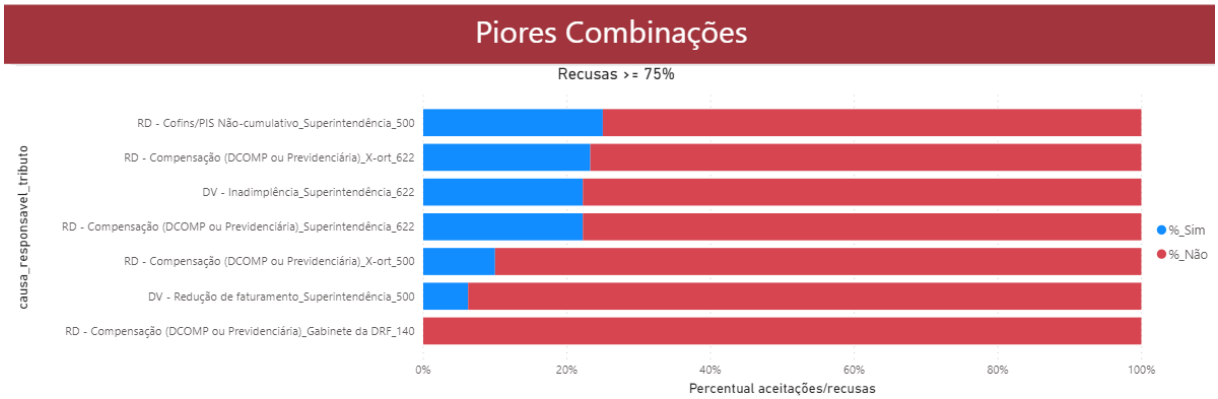
Registro	Frequência	Nível de aceitação
CA nível 3 140	2º mais frequente	2º mais aceito
Responsável pelo tratamento X-pac	2º mais frequente	3º mais aceito
Causa da Distorção RD - Cofins/PIS Não Cumulativo	3º mais frequente	4º mais aceito. Alto quando combinado com X-fis ou X-pac e CA 500 ou combinado com CA 140
Causa da Distorção DV - Inadimplência	2º mais frequente	Alto quando combinado com Gabinete de DRF e CA 622

Pontos Críticos Encontrados

Registros com maior número de ocorrências	Nível de aceitação
Responsável pelo Tratamento X-ort	Menos aceito
Causa da Distorção RD - Compensação (DCOMP ou Previdenciária)	2º menos aceito
CA nível 3 622	2º menos aceito

Melhores e piores combinações de atributos com relação à aceitação:

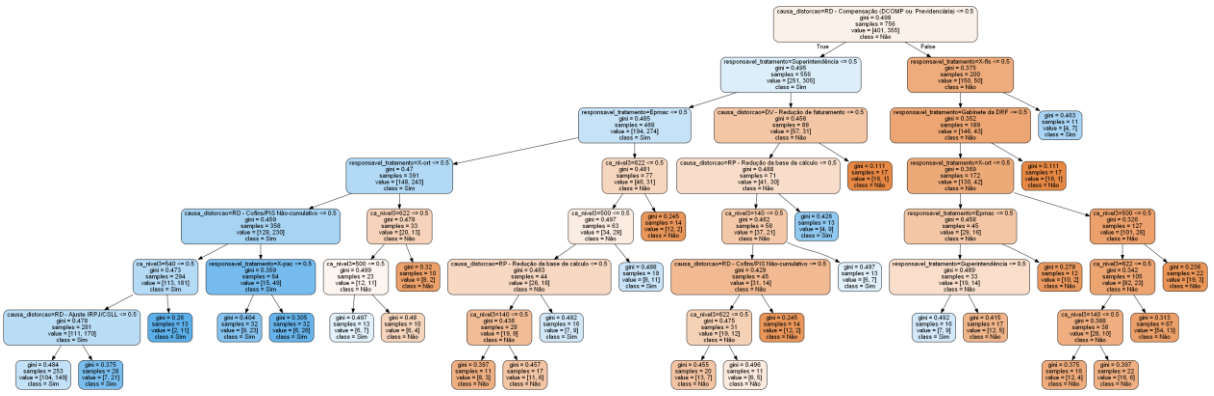




6.2. Resultados da criação dos Modelos de Machine Learning

Na etapa de criação e avaliação de modelos de Machine Learning, o algoritmo mais bem avaliado foi o Decision Tree Classifier. Isso se mostrou conveniente para fins práticos, pois se pretende que seja possível que as pessoas que trabalham com as análises e encaminhamentos possam consultar, preferencialmente de forma visual, os melhores caminhos e escolhas que levam a melhores ou piores níveis de aceitação.

Através da funcionalidade do pydotplus, foi gerada a seguinte árvore de decisão, disponível no repositório como o arquivo “Árvore.png”:



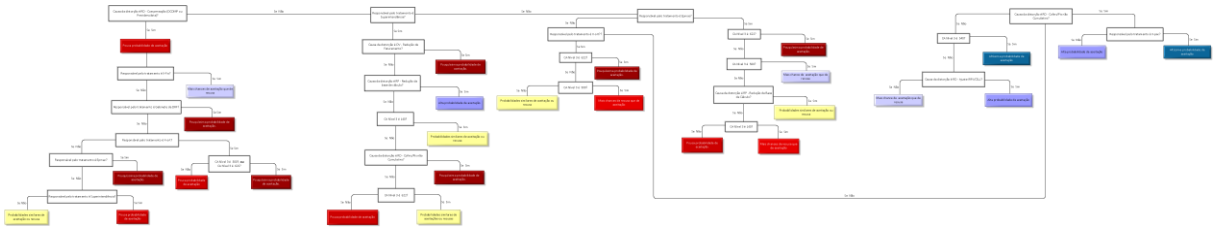
6.2.1. Árvore de Decisão Simplificada

Para fins de consulta pelos usuários comuns, que não compreendem os conceitos relacionados a Machine Learning, essa Decision Tree pode parecer complexa e de compreensão mais difícil. Por esse motivo, para fins pragmáticos de ser consultada por todos que possam se interessar, bem como visando a uma melhor compreensão da árvore gerada pelo pydotplus, baseando-se nesta última, foi criada uma árvore de decisão simplificada com o auxílio da ferramenta Aris Express que demonstraremos abaixo.

Dependendo da proporção de Sim no nó folha da árvore original acima a probabilidade de aceitação será considerada nessa árvore como:

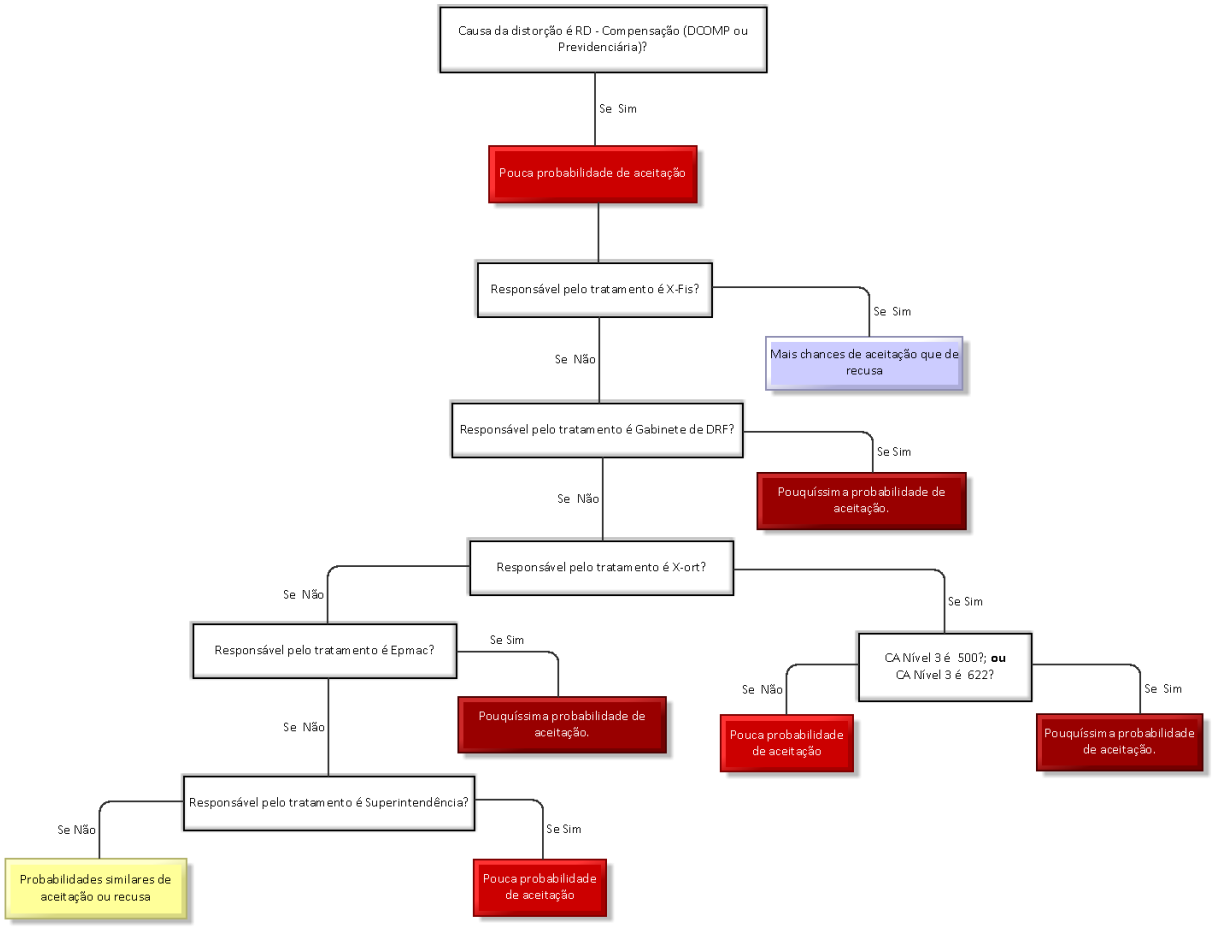
Representação do nó	Faixa de proporção de aceitação	Probabilidade de aceitação
Pouquíssima probabilidade de aceitação.	<= 21%	Pouquíssima probabilidade de aceitação
Pouca probabilidade de aceitação	> 21% e <= 35%	Pouca probabilidade de aceitação
Mais chances de recusa que de aceitação	> 35% e < 42,5%	Mais chances de recusa que de aceitação
Probabilidades similares de aceitação ou recusa	>= 42,5% e <=57,5%	Probabilidades similares de aceitação e recusa
Mais chance de aceitação que de recusa	> 57,5% e <= 65%	Mais chances de aceitação que de recusa
Alta probabilidade de aceitação	> 65 % e <= 80%	Alta probabilidade de aceitação
Altíssima probabilidade de aceitação	> 80%	Altíssima probabilidade de aceitação

Pode-se verificar a árvore simplificada completa resultante abaixo (detalhamento a seguir):

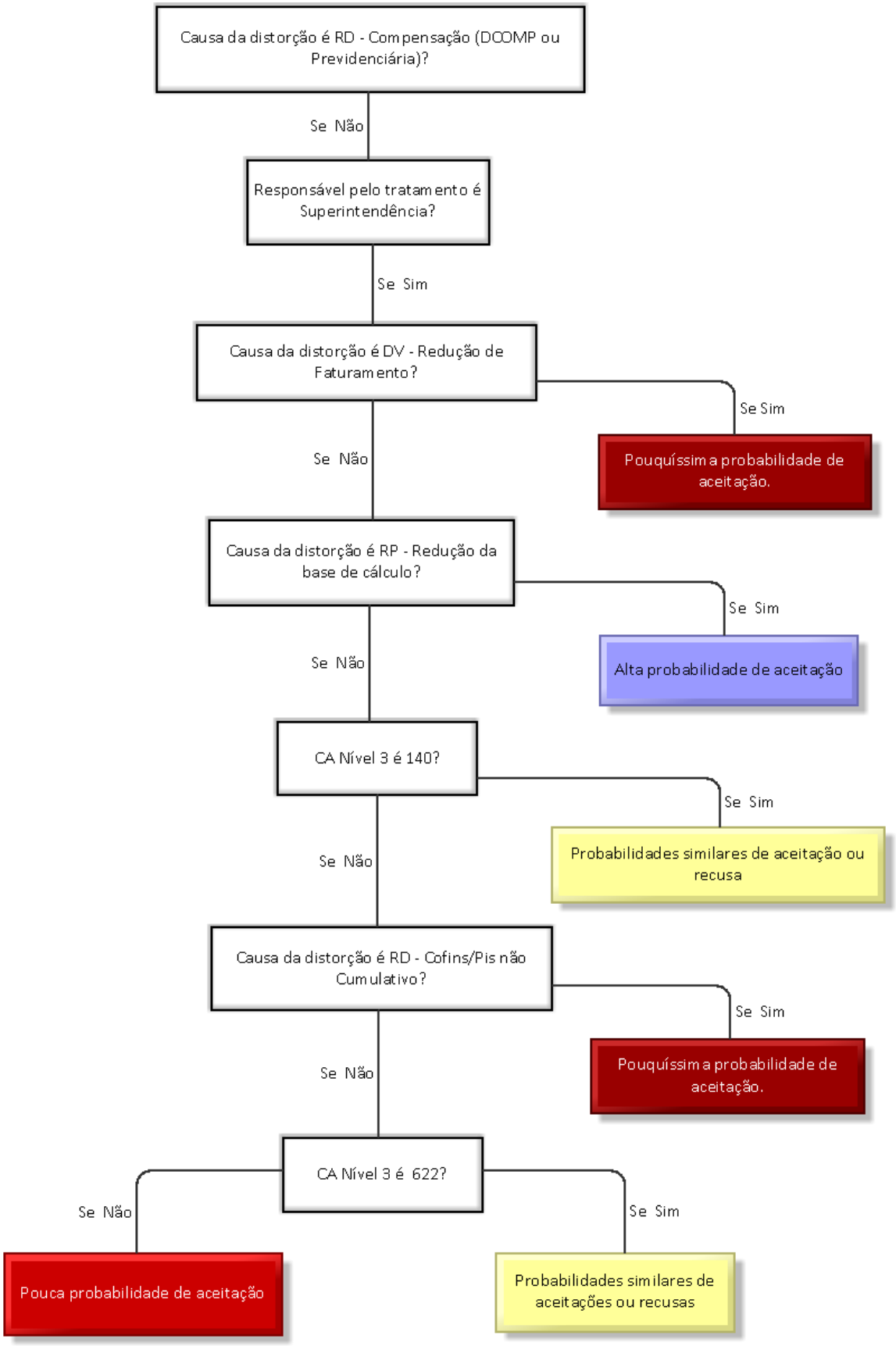


Para uma melhor visualização, a árvore acima foi dividida em quatro partes, representando-se na sequência as sub-árvores vistas da esquerda para a direita:

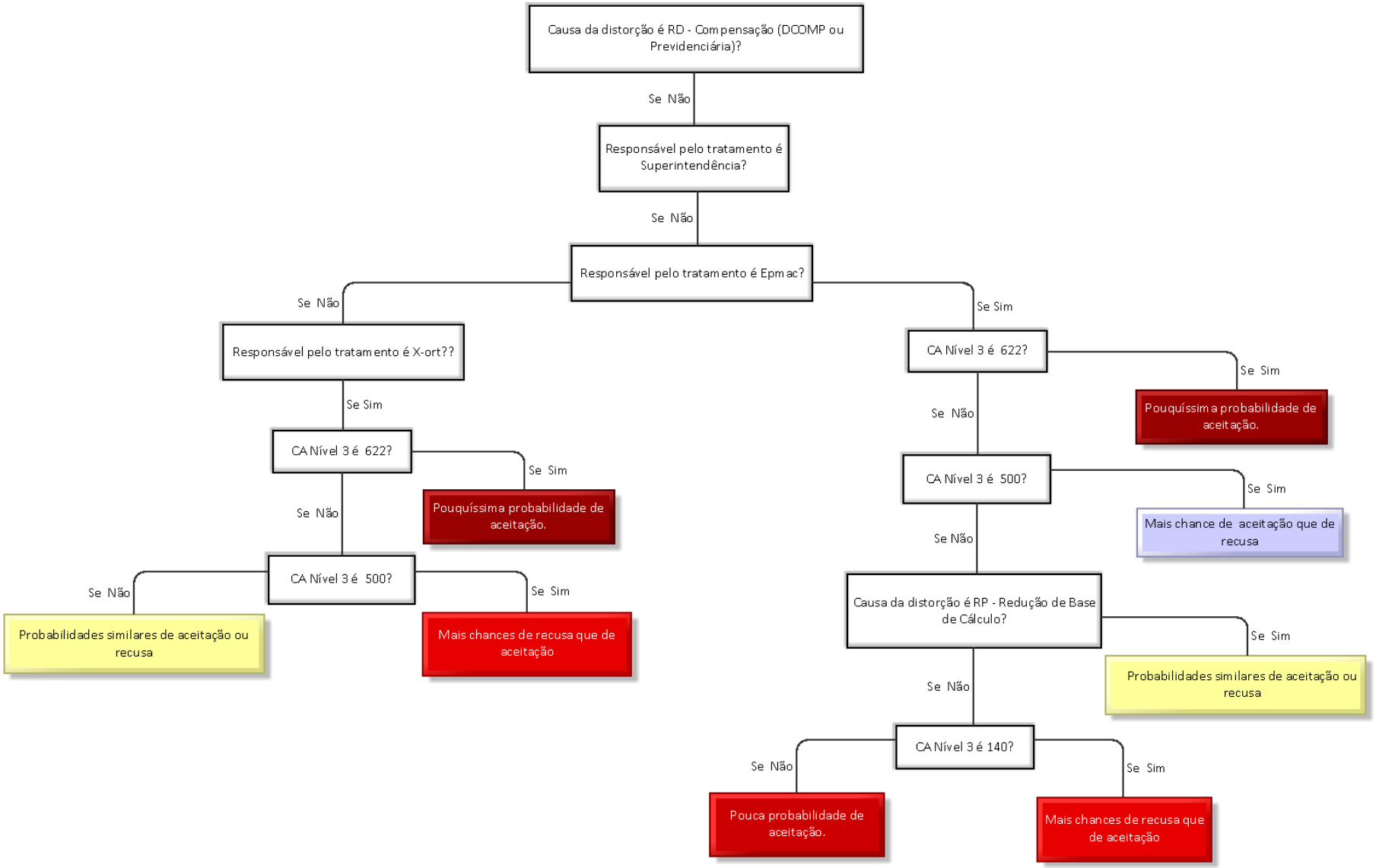
Parte 1:


Causa da distorção é RD - Compensação (DCOMP ou Previdenciária)?
Se Sim
Pouca probabilidade de aceitação
Responsável pelo tratamento é X-Fis?
Se Sim
Mais chances de aceitação que de recusa
Se Não
Responsável pelo tratamento é Gabinete de DRF?
Se Sim
Pouquíssima probabilidade de aceitação.
Se Não
Responsável pelo tratamento é X-ort?
Se Não
Se Sim
Responsável pelo tratamento é Epmac?
Se Sim
Pouquíssima probabilidade de aceitação.
CA Nível 3 é 500?; ou CA Nível 3 é 622?
Se Não
Pouca probabilidade de aceitação
Se Sim
Pouquíssima probabilidade de aceitação.
Se Não
Responsável pelo tratamento é Superintendência?
Se Não
Probabilidades similares de aceitação ou recusa
Se Sim
Pouca probabilidade de aceitação

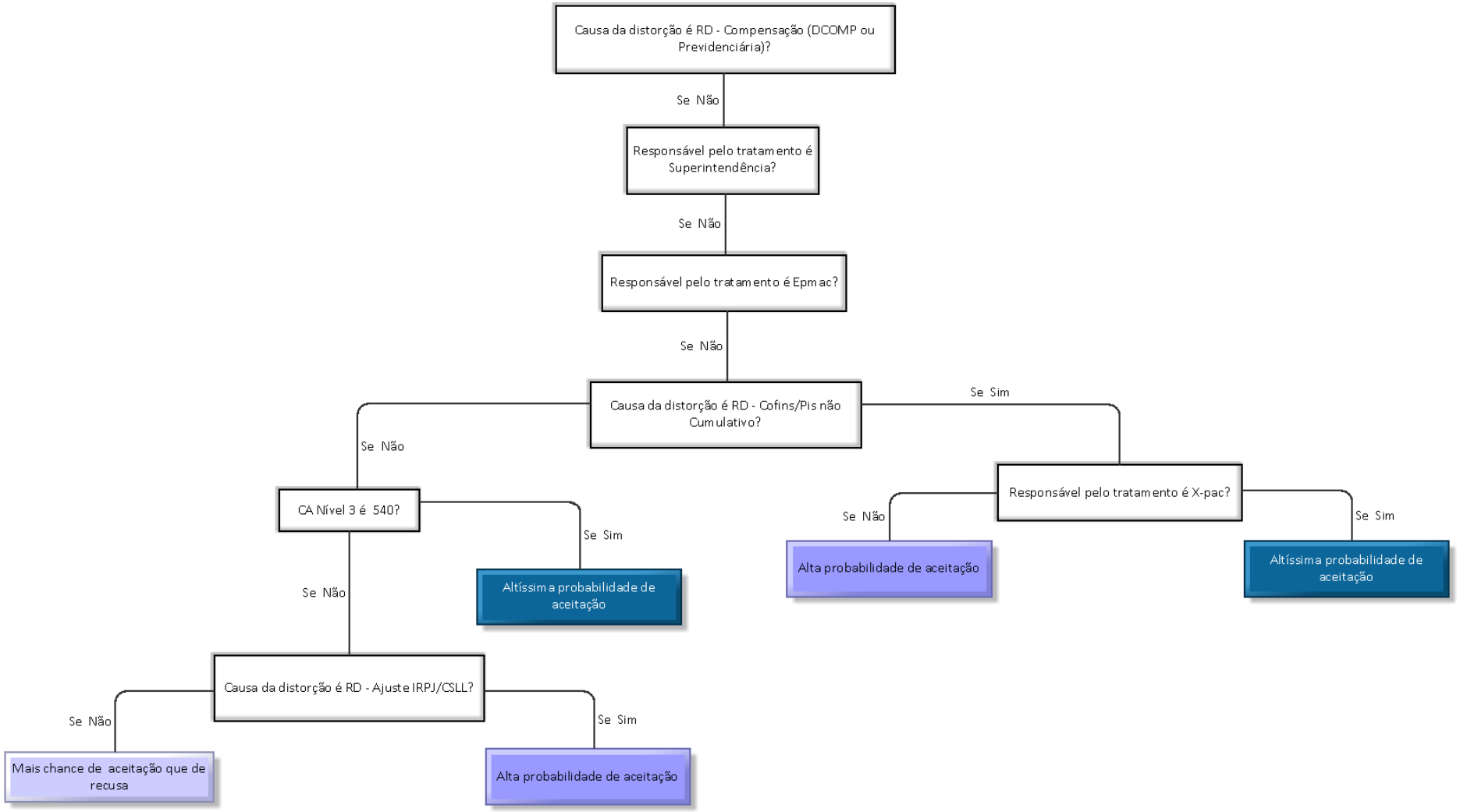
Parte 2:


Causa da distorção é RD - Compensação (DCOMP ou Previdenciária)?
Se Não
Responsável pelo tratamento é Superintendência?
Se Sim
Causa da distorção é DV - Redução de Faturamento?
Se Sim
Pouquíssima probabilidade de aceitação.
Se Não
Causa da distorção é RP - Redução da base de cálculo?
Se Sim
Alta probabilidade de aceitação
Se Não
CA Nível 3 é 140?
Se Sim
Probabilidades similares de aceitação ou recusa
Se Não
Causa da distorção é RD - Cofins/Pis não Cumulativo?
Se Sim
Pouquíssima probabilidade de aceitação.
Se Não
CA Nível 3 é 622?
Se Não
Pouca probabilidade de aceitação
Se Sim
Probabilidades similares de aceitações ou recusas

Parte 3:


Causa da distorção é RD - Compensação (DCOMP ou Previdenciária)?
Se Não
Responsável pelo tratamento é Superintendência?
Se Não
Responsável pelo tratamento é Epmac?
Se Não
Responsável pelo tratamento é X-ort??
Se Sim
CA Nível 3 é 622?
Se Sim
Pouquíssima probabilidade de aceitação.
Se Não
CA Nível 3 é 500?
Se Não
Probabilidades similares de aceitação ou recusa
Se Sim
Mais chances de recusa que de aceitação
Se Sim
CA Nível 3 é 622?
Se Sim
Pouquíssima probabilidade de aceitação.
Se Não
CA Nível 3 é 500?
Se Sim
Mais chance de aceitação que de recusa
Se Não
Causa da distorção é RP - Redução de Base de Cálculo?
Se Sim
Probabilidades similares de aceitação ou recusa
Se Não
CA Nível 3 é 140?
Se Não
Pouca probabilidade de aceitação.
Se Sim
Mais chances de recusa que de aceitação

Parte 4:


Causa da distorção é RD - Compensação (DCOMP ou Previdenciária)?
Se Não
Responsável pelo tratamento é Superintendência?
Se Não
Responsável pelo tratamento é Epmac?
Se Não
Causa da distorção é RD - Cofins/Pis não Cumulativo?
Se Sim
Se Não
CA Nível 3 é 540?
Se Sim
Altíssima probabilidade de aceitação
Se Não
Causa da distorção é RD - Ajuste IRPJ/CSLL?
Se Não
Mais chance de aceitação que de recusa
Se Sim
Alta probabilidade de aceitação
Responsável pelo tratamento é X-pac?
Se Não
Alta probabilidade de aceitação
Se Sim
Altíssima probabilidade de aceitação

A árvore de decisão simplificada foi um resultado que pode ter grande utilidade, e permitirá auxiliar na tomada de decisões dos melhores caminhos para aceitações através de um recurso intuitivo e visual, podendo ser facilmente interpretada por qualquer pessoa que dela necessitar.

6.3. Conclusão

Espera-se que os responsáveis pelos encaminhamentos a outros processos de trabalho possam se basear nesse projeto, seus gráficos, dados e árvores de decisão para que realmente se alcancem cada vez melhores níveis de aceitação.

Também será interessante examinar os pontos fracos e casos mais recusados para que se possa fazer uma análise de como pode haver melhoria. Esses achados negativos podem significar pistas para que haja, por exemplo, ações de capacitação voltadas a lidar melhor com temas que podem estar gerando dificuldades para os responsáveis pelo tratamento, ou alguma forma de premiá-los ou persuadi-los quando do encaminhamento de análises com características menos aceitas atualmente.

Por fim, com o passar do tempo, os datasets aumentarão e poderão ser feitas novas análises com a utilização de novos dados, o que poderá levar a novas classificações.

7. Links

Aqui você deve disponibilizar os links para o vídeo com sua apresentação de 5 minutos e para o repositório contendo os dados utilizados no projeto, scripts criados, etc.

Link para o vídeo: https://www.youtube.com/watch?v=Z9ZK9HMde00

Link para o repositório: https://github.com/daniel-casmar/TCC

APÊNDICE

Programação/Scripts

Script usado para os describes do Dataset Inicial:

```
import pandas as pd
```

1 - Dataset Inicial

```
aceitacoes = pd.read_excel('Aceitações_final.xlsx')
```

```
aceitacoes.shape
```

```
(1486, 4)
```

```
aceitacoes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1486 entries, 0 to 1485
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   num_analise    1486 non-null   int64 
 1   ano_apuracao   1486 non-null   int64 
 2   rf             1486 non-null   object
 3   aceitacao      1486 non-null   object
dtypes: int64(2), object(2)
memory usage: 46.6+ KB
```

```
aceitacoes = aceitacoes.astype(str)
aceitacoes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1486 entries, 0 to 1485
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   num_analise    1486 non-null   object
 1   ano_apuracao   1486 non-null   object
 2   rf             1486 non-null   object
 3   aceitacao      1486 non-null   object
dtypes: object(4)
memory usage: 46.6+ KB
```

```
aceitacoes.head()
```

	num_analise	ano_apuracao	rf	aceitacao
0	10785	2016	RF04	Não
1	33199	2016	RF01	Não
2	33239	2016	RF01	Não
3	41499	2016	RF01	Não
4	55554	2016	RF01	Não

1.1 - Describe simples

```
aceitacoes.iloc[:,1:].describe()
```

1.2 - Describe por Ano de Apuração

```
aceitacoes[['ano_apuracao', 'aceitacao']].groupby(['ano_apuracao'], as_index=False).count().describe()
```

	aceitacao
count	5.000000
mean	297.200000
std	151.283509
min	148.000000
25%	194.000000
50%	295.000000
75%	310.000000
max	539.000000

1.3 - Describe por Regiões Fiscais

```
aceitacoes[['rf', 'aceitacao']].groupby(['rf'], as_index=False).count().describe()
```

	aceitacao
count	10.000000
mean	148.600000
std	88.589189
min	58.000000
25%	97.250000
50%	112.500000
75%	162.250000
max	323.000000

SQL de tratamento para geração do Dataset para Modelagem:

```
with base as
(select
t2.nb_mcpj_anal_causa_distorcao as causa_distorcao,
t3.nb_mcpj_anal_setor_resp as responsavel_tratamento,
case
    when (t5.cd_ca_nivel3 != "NI" and t5.cd_ca_nivel3 != "650") then t5.cd_ca_nivel3
    when (t7.nm_anal_obs_diagnostico ilike "%IRPJ%" or t7.nm_anal_obs_diagnostico ilike "%ECF%" or t7.nm_anal_obs_diagnostico ilike "%Lucro real%" or t7.nm_anal_obs_diagnostico ilike "%Lalur%" or t7.nm_anal_obs_diagnostico ilike "%IRRF%" or t7.nm_anal_obs_diagnostico ilike "%DIPJ%" or t7.nm_anal_obs_diagnostico ilike "%perdas não técnicas%" or
```

```
t7.nm_anal_obs_diagnostico ilike "%perdas não-técnicas%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%IRPJ%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%balancete%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%lucro%") then "140"
   when (t7.nm_anal_obs_diagnostico ilike "%previd%" or
t7.nm_anal_obs_diagnostico ilike "%GFIP%" or t7.nm_anal_obs_diagnostico ilike
"%GPS%" or t7.nm_anal_obs_diagnostico ilike "%CPRB%" or
t7.nm_anal_obs_diagnostico ilike "%Dacon%" or t2.nb_mcpj_anal_causa_distorcao
ilike "%GPS%") then "622"
   when (t7.nm_anal_obs_diagnostico ilike "%Cofins%" or
t7.nm_anal_obs_diagnostico ilike "%EFD%" or t7.nm_anal_obs_diagnostico ilike
"%Sped Contribuições%" or t7.nm_anal_obs_diagnostico ilike "2172" or
t2.nb_mcpj_anal_causa_distorcao ilike "%Cofins%") then "500"
   when t7.nm_anal_obs_diagnostico ilike "%Pagamento Unificado%" then "644"
   when t7.nm_anal_obs_diagnostico ilike "%CSLL%" then "580"
      when (t7.nm_anal_obs_diagnostico ilike "%Pis%" or
t7.nm_anal_obs_diagnostico ilike "%Pasep%" or t7.nm_anal_obs_diagnostico ilike
"%8109%") then "540"
   when (t7.nm_anal_obs_diagnostico ilike "%IPI%" or
t2.nb_mcpj_anal_causa_distorcao ilike "%IPI%") then "060"
   when t7.nm_anal_obs_diagnostico ilike "%IOF%" then "360"
   when t7.nm_anal_obs_diagnostico ilike "%CIDE%" then "612"
   when t7.nm_anal_obs_diagnostico ilike "%Fundaf%" then  "620"
   when (t7.nm_anal_obs_diagnostico = "Não informado" or  t5.cd_ca_nivel3 = "NI")
then "Não informado"
   else "Outros" end as ca_nivel3,
      t6.aceitacao as aceitacao
from mcpj.wf_mcpj_anals as t1
   join mcpj.wd_mcpj_anal_causa_distorcaos as t2
     on t1.nr_mcpj_anal_causa_distorcao = t2.nr_mcpj_anal_causa_distorcao
   join mcpj.wd_mcpj_anal_setor_resps as t3
     on t1.nr_mcpj_anal_setor_resp = t3.nr_mcpj_anal_setor_resp
   join dime.wd_rc_ca_nivel6 as t4
     on t1.nr_mcpj_csel_ca_n6_h_crit_sel = t4.nr_ca_nivel6
```

```
    join dime.wd_rc_ca_nivel3 as t5
      on t4.nr_ca_nivel3 = t5.nr_ca_nivel3
    join u01406263605.aceitacoes_v2 as t6
      on t1.dd_anal_num_analise = t6.num_analise
    join mcpj.wd_mcpj_anals as t7
      on (t1.dd_anal_num_analise = t7.dd_anal_num_analise and t1.nr_mcpj_anal = t7.nr_mcpj_anal)
where t1.dd_anal_num_analise in (select num_analise from u01406263605.aceitacoes_v2) )

select
causa_distorcao,
responsavel_tratamento,
ca_nivel3,
aceitacao
from base
where causa_distorcao not in ("XX - Causa não identificada", "XX - Outras (descrever em observação)")
and responsavel_tratamento not in ("Não informado")
and ca_nivel3 not in ("Não informado", "Outros")
```

Script usado para os describes do Dataset Inicial:

```
In [3]: import pandas as pd
```

2 - Dataset para Modelagem

```
dataset = pd.read_excel('DATASET_PARA_CLASSIFICACAO.xlsx')
```

```
dataset
```

	causa_distorcao	responsavel_tratamento	ca_nivel3	aceitacao
0	RD - Compensação (DCOMP ou Previdenciária)	Gabinete da DRF	580	Não
1	RD - Compensação (DCOMP ou Previdenciária)	Gabinete da DRF	140	Não
2	RD - Compensação (DCOMP ou Previdenciária)	Gabinete da DRF	140	Não
3	RD - Compensação (DCOMP ou Previdenciária)	Gabinete da DRF	140	Não
4	RD - Compensação (DCOMP ou Previdenciária)	Gabinete da DRF	140	Não
...	...	...	...	...
1004	RP - Redução da base de cálculo	Superintendência	140	Sim
1005	DV - Indício de possível irregularidade tribut...	X-fis	540	Sim
1006	RD - Compensação (DCOMP ou Previdenciária)	X-pac	500	Sim
1007	RD - Compensação (DCOMP ou Previdenciária)	X-pac	540	Sim
1008	RP - Redução da base de cálculo	X-fis	140	Sim

1009 rows × 4 columns

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1009 entries, 0 to 1008
Data columns (total 4 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   causa_distorcao        1009 non-null   object
 1   responsavel_tratamento 1009 non-null   object
 2   ca_nivel3              1009 non-null   int64 
 3   aceitacao              1009 non-null   object
dtypes: int64(1), object(3)
memory usage: 31.7+ KB
```

```
dataset = dataset.astype(str)
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1009 entries, 0 to 1008
Data columns (total 4 columns):
 #   Column                 Non-Null Count  Dtype 
---  ------                 --------------  ----- 
 0   causa_distorcao        1009 non-null   object
 1   responsavel_tratamento 1009 non-null   object
 2   ca_nivel3              1009 non-null   object
 3   aceitacao              1009 non-null   object
dtypes: object(4)
memory usage: 31.7+ KB
```

2.1 - Describe simples

```
dataset.describe()
```

	causa_distorcao	responsavel_tratamento	ca_nivel3	aceitacao
count	1009	1009	1009	1009
unique	38	11	11	2
top	RD - Compensação (DCOMP ou Previdenciária)	X-ort	622	Não
freq	270	221	305	528

2.2 - Describe por Causa da Distorção

```
dataset[['causa_distorcao', 'aceitacao']].groupby(['causa_distorcao'], as_index= False).count().describe()
```

	aceitacao
count	38.000000
mean	26.552632
std	55.494678
min	1.000000
25%	2.000000
50%	4.500000
75%	17.750000
max	270.000000

2.3 - Describe por Responsável pelo Tratamento

```
dataset[['responsavel_tratamento', 'aceitacao']].groupby(['responsavel_tratamento'], as_index= False).count().describe()
```

	aceitacao
count	11.000000
mean	91.727273
std	80.772633
min	1.000000
25%	6.000000
50%	109.000000
75%	142.000000
max	221.000000

2.4 - Describe por CA Nível 3

```
dataset[['ca_nivel3', 'aceitacao']].groupby(['ca_nivel3'], as_index= False).count().describe()
```

	aceitacao
count	11.000000
mean	91.727273
std	120.836328
min	1.000000
25%	3.000000
50%	37.000000
75%	164.500000
max	305.000000

Script usado para Modelos de Machine Learning:

1 - Importação de bibliotecas

```
# Retirar as aspas triplas para instalar as bibliotecas necessárias, se não instaladas anteriormente
'''!pip install pydotplus
!pip install dtreeviz'''
```

```
'!pip install pydotplus\n!pip install dtreeviz'
```

```
import pandas as pd
import numpy as np
import pydotplus
from IPython.display import Image
from sklearn import datasets, tree
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.feature_extraction import DictVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from dtreeviz.trees import *
```

2 - Carga dos dados e particionamento das bases de treinamento e teste

```
aceitacoes = pd.read_excel('DATASET_PARA_CLASSIFICACAO.xlsx', sheet_name=0)
print("\nDimensões: {0}".format(aceitacoes.shape))
print("\nCampos: {0}".format(aceitacoes.keys()))


aceit_format = aceitacoes.astype(str)
print(aceit_format.describe(), sep='\n')

le = LabelEncoder()
X_dict = aceit_format.iloc[:,0:(aceit_format.shape[1] - 1)].T.to_dict().values()
vect = DictVectorizer(sparse=False)
X = vect.fit_transform(X_dict)

y = le.fit_transform(aceit_format.iloc[:,(aceit_format.shape[1] - 1)])


# Particiona a base de dados utilizando 25% para teste e 75% para treinamento
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size=0.25)

# Exibe o dado convertido em dicionario.
print("Atributos:", X_dict)

# Exibe a estrutura do dado convertido em binário.
print("Shape do dado de treinamento: {0}".format(X_train.shape))

print("Labels:", y_train)
```

3 - Modelos de Machine Learning

3.1 - Decision Tree Classifier (Árvore de Decisão)

```
# Melhor avaliado
aceit_tree = DecisionTreeClassifier(random_state=42, criterion='gini', max_depth = 7, min_samples_leaf = 9)
aceit_tree = aceit_tree.fit(X_train, y_train)
print("Acurácia (base de treinamento):", aceit_tree.score(X_train, y_train))

print('='*60)

print('       Acurácia de Previsão e Classification Report\n')
y_pred = aceit_tree.predict(X_test)
print("Acurácia de previsão:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred, target_names=["Sim", "Não"]))

print('='*60)

print('      Matriz de Confusão\n')
cnf_matrix = confusion_matrix(y_test, y_pred)
cnf_table = pd.DataFrame(data=cnf_matrix, index=["Sim", "Não"], columns=["Sim (prev)", "Não (prev)"])
print(cnf_table)
```

```
scores_tree = cross_val_score(aceit_tree, X_train, y_train, scoring='accuracy', cv=4)
print('Acurácia por Cross Validation\n')
print(scores_tree.mean())
```

3.2 - Random Forest

```
aceit_forest = RandomForestClassifier(random_state = 42, criterion='gini', max_depth = 7, min_samples_leaf = 9, n_estimators=17,
                                      n_jobs=-1)
aceit_forest = aceit_forest.fit(X_train, y_train)
print("Acurácia (base de treinamento):", aceit_forest.score(X_train, y_train))

print('='*60)

print('       Acurácia de Previsão e Classification Report\n')
y_pred = aceit_forest.predict(X_test)
print("Acurácia de previsão:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred, target_names=["Sim", "Não"]))

print('='*60)

print('      Matriz de Confusão\n')
cnf_matrix = confusion_matrix(y_test, y_pred)
cnf_table = pd.DataFrame(data=cnf_matrix, index=["Sim", "Não"], columns=["Sim (prev)", "Não (prev)"])
print(cnf_table)
```

```
scores_forest = cross_val_score(aceit_forest, X_train, y_train, scoring='accuracy', cv=4)
print('Acurácia por Cross Validation\n')
print(scores_forest.mean())
```

3.3 - Adaboost

```
aceit_adab = AdaBoostClassifier(base_estimator = DecisionTreeClassifier(random_state=42, criterion='gini', max_depth = 7,
                                                 min_samples_leaf = 9), n_estimators=10, random_state=42, algorithm='SAMME')
aceit_adab = aceit_adab.fit(X_train, y_train)
print("Acurácia (base de treinamento):", aceit_adab.score(X_train, y_train))

print('='*60)

print('       Acurácia de Previsão e Classification Report\n')
y_pred = aceit_adab.predict(X_test)
print("Acurácia de previsão:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred, target_names=["Sim", "Não"]))

print('='*60)

print('      Matriz de Confusão\n')
cnf_matrix = confusion_matrix(y_test, y_pred)
cnf_table = pd.DataFrame(data=cnf_matrix, index=["Sim", "Não"], columns=["Sim (prev)", "Não (prev)"])
print(cnf_table)
```

```
scores_adab = cross_val_score(aceit_adab, X_train, y_train, scoring='accuracy', cv=4)
print('Acurácia por Cross Validation\n')
print(scores_adab.mean())
```

3.4 - Nayve Bayes

```
aceit_gnb = GaussianNB()
aceit_gnb = aceit_gnb.fit(X_train, y_train)
print("Acurácia (base de treinamento):", aceit_gnb.score(X_train, y_train))

print('='*60)

print('       Acurácia de Previsão e Classification Report\n')
y_pred = aceit_gnb.predict(X_test)
print("Acurácia de previsão:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred, target_names=["Sim", "Não"]))

print('='*60)

print('      Matriz de Confusão\n')
cnf_matrix = confusion_matrix(y_test, y_pred)
cnf_table = pd.DataFrame(data=cnf_matrix, index=["Sim", "Não"], columns=["Sim (prev)", "Não (prev)"])
print(cnf_table)
```

```
scores_nb = cross_val_score(aceit_gnb, X_train, y_train, scoring='accuracy', cv=4)
print('Acurácia por Cross Validation\n')
print(scores_nb.mean())
```

4 - Exibição da árvore de decisão

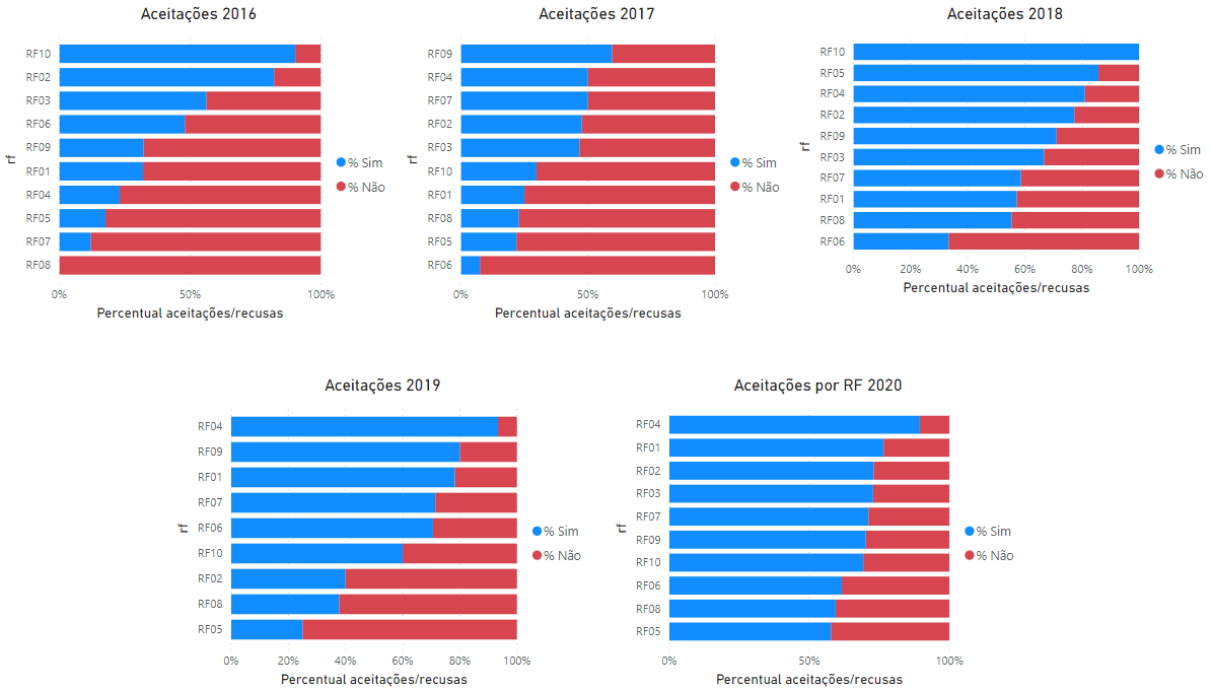
```
import os

os.environ['PATH'] = os.environ['PATH']+';'+r"C:\Users\Daniel\Anaconda3\Library\bin\graphviz"
```

```
# Create DOT data
dot_data = tree.export_graphviz(aceit_tree, out_file=None,
                                proportion=False,
                                rounded =True,
                                filled=True,
                                feature_names=vect.feature_names_,
                                class_names=["Não", "Sim"])

# Draw graph
graph = pydotplus.graph_from_dot_data(dot_data)

# Show graph
Image(graph.create_png())
```

```
viz = dtreeviz(aceit_tree,
               X_train,
               y_train,
               target_name="Aceitacao",
               feature_names = vect.feature_names_,
               class_names = ["Sim", "Não"])

viz.view()
```

Gráficos

Aceitações por Regiões Fiscais por Ano de Apuração:



Árvore de Decisão gerada pelo DtreeViz: