

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

ALESSANDRO ZABOTTO

**Análise dos dados da Central de Atendimento
ao Cliente para melhoria do produto e serviço**

Paulínia / SP

2023

ALESSANDRO ZABOTTO

**Análise dos dados da Central de Atendimento
ao Cliente para melhoria do produto e serviço**

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.

Paulínia / SP

2023

Sumário

| | |
|--|-----------|
| 1. Introdução | 9 |
| 1.1. Contextualização | 9 |
| 1.1.1 O que é a CAC..... | 9 |
| 1.1.2 O que é Customer Experience (CX) | 11 |
| 1.2.3 Jornada do cliente | 13 |
| 1.2. O problema proposto | 14 |
| 1.3. Objetivos..... | 15 |
| 2. Coleta de Dados..... | 16 |
| 3. Processamento/Tratamento de Dados | 19 |
| 3.1 Pré-processamento dos dados..... | 19 |
| 3.1.1 Dataset CAC..... | 19 |
| 3.1.1.1 Preparação dos dados do dataset CAC no formato .csv | 19 |
| 3.1.1.2 Leitura e análise do dataset CAC.csv..... | 20 |
| 3.1.2 Dataset RVAT..... | 26 |
| 3.1.2.1 Preparação dos dados do dataset RVAT no formato .csv | 26 |
| 3.1.2.2 Leitura e análise do dataset RVAT.csv..... | 26 |
| 4. Análise e Exploração dos Dados | 28 |
| 4.1 – Dataset CAC..... | 28 |
| 4.2 – Dataset RVAT..... | 35 |
| 4.3 – Relacionando os datasets CAC e RVAT | 37 |
| 5. Criação de Modelos de Machine Learning..... | 43 |
| 5.1 – Bag Of Words..... | 43 |
| 5.2 - Term Frequency - Inverse Document Frequency (TF-IDF) | 43 |
| 5.3 – Implementação NLP | 44 |
| 5.3.1 – Seleção dos dados | 44 |
| 5.3.2 – Pré-processamento do texto..... | 45 |
| 5.3.1.1 Etapas do pré-processamento do texto..... | 45 |
| a) Expressões regulares..... | 45 |
| b) Tokenização..... | 45 |

| | |
|---|-----------|
| c) Stopwords..... | 45 |
| d) Lematização | 46 |
| 5.3.1.2 Geração do BoW e TF-IDF | 49 |
| 5.3.1.3 Análise e aplicação da redução da dimensionalidade..... | 52 |
| 5.3.1.4 K-Means..... | 55 |
| 5.3.1.4.1 Método curva de cotovelo | 55 |
| 5.3.1.4.2 Silhouette | 56 |
| 5.3.1.4.3 Aplicação do K-Means | 57 |
| 6. Interpretação dos Resultados | 62 |
| 7. Apresentação dos Resultados | 64 |
| The machine learning CANVAS | 70 |
| 8. Links | 72 |
| Referências | 73 |
| Apêndice | 74 |

Índice de ilustrações e figuras

| | |
|--|----|
| Figura 1 - Percentual de utilização dos canais de contato com o CAC..... | 9 |
| Figura 2 - Perguntas feitas na pesquisa NPS | 12 |
| Figura 3 - Classificação do NPS..... | 12 |
| Figura 4 - Etapas da jornada do cliente na indústria automobilística..... | 13 |
| Figura 5 - 5W para auxílio para interpretação do problema proposto..... | 14 |
| Figura 6 - Descaracterização dos dados (sigilo e LGPD)..... | 19 |
| Figura 7 - Exportação dos dados para formato .csv..... | 20 |
| Figura 8 - Instalação e importação das bibliotecas..... | 21 |
| Figura 9 - Desabilitando as mensagens de erros referentes a “DeprecationWarning” | 21 |
| Figura 10 - Leitura do dataset CAC..... | 21 |
| Figura 11 - Leitura dos 5 primeiros registros do dataset CAC..... | 22 |
| Figura 12 - Leitura dos 5 últimos registros do dataset CAC | 22 |
| Figura 13 - Análise de registros duplicados..... | 22 |
| Figura 14 - Exclusão da coluna "NrEvento"..... | 23 |
| Figura 15 - Calculando o tamanho das colunas "Descricao" e "ExpectativaCliente" | 23 |
| Figura 16 - Concatenando as colunas "Descricao" e "ExpectativaCliente" | 23 |
| Figura 17 - Exclusão das colunas "ExpectativaCliente" e "TamExpectativaCliente" | 24 |
| Figura 18 - Análise dos 5 primeiros registros após exclusão das colunas..... | 24 |
| Figura 19 - Tratamento da coluna "Data" | 24 |
| Figura 20 - Conversão do tipo da coluna "Data" para datetime | 25 |
| Figura 21 - Inclusão das colunas "Mes" e "Ano" | 25 |
| Figura 22 - Leitura dos 5 primeiros registros do dataset CAC com as colunas mês e ano..... | 25 |
| Figura 23 - Tratamento dos dados do dataset “RVAT.xlsx” | 26 |
| Figura 24 - leitura do dataset RVAT | 26 |
| Figura 25 - Amostragem dos registros do dataset RVAT..... | 27 |
| Figura 26 - Visualizando todas as UF que constam no dataset RVAT | 27 |

| | |
|--|----|
| Figura 27 - Segmentos que constam no dataset CAC | 28 |
| Figura 28 - Dataset CAC após aplicar o filtro por "Truck" e "Bus" | 29 |
| Figura 29 – Código p/ plotagem do gráfico comparativo atendimentos CAC / Ano..... | 29 |
| Figura 30 - Gráfico comparativo de atendimento CAC / Ano | 30 |
| Figura 31 - Primeira e última data de atendimento CAC constante no dataset CAC..... | 30 |
| Figura 32 - Visualização dos dados por classificação do atendimento CAC..... | 30 |
| Figura 33 - Gráfico dos atendimentos CAC por ano e classificação | 31 |
| Figura 34 – Distribuição e concentração do "TamDescricao" | 32 |
| Figura 35 - Distribuição e concentração do "TamDescricao" através do diagrama de caixas | 32 |
| Figura 36 - Estatística do "TamDescricao" | 33 |
| Figura 37 – Amostra de registros do dataset CAC sem "Descricao" e/ou "ExpectativaCliente" | 33 |
| Figura 38 - Registro do dataset CAC com maior tamanho (qtde. de caracteres) da coluna "Descricao" | 34 |
| Figura 39 - Informações do dataset RVAT | 35 |
| Figura 40 - Código para geração do gráfico de RVAT por UF | 35 |
| Figura 41 - Distribuição da RVAT por UF | 35 |
| Figura 42 - Amostragem do dataset RVAT | 36 |
| Figura 43 - Frota circulante x RVAT | 36 |
| Figura 44 - RVAT por UF | 37 |
| Figura 45 - Relacionamento entre os datasets RVAT e CAC..... | 37 |
| Figura 46 - Verificando qtde de registros com “GSSN” nulo | 38 |
| Figura 47 - Dados gerais após relacionamento dos datasets RVAT e CAC..... | 39 |
| Figura 48 - Amostragem dos dados após o relacionamento dos datasets RVAT e CAC | 39 |
| Figura 49 - Código para plotagem do gráfico atendimento CAC por UF..... | 40 |
| Figura 50 - Gráfico de atendimento CAC por UF..... | 40 |
| Figura 51 - Código para plotagem do mapa de calor através da biblioteca "Folium" | 41 |
| Figura 52 - Mapa de calor atendimento CAC nacional..... | 42 |
| Figura 53 - Cálculo do TF-IDF..... | 44 |

| | |
|--|----|
| Figura 54 - Fluxo tratamento texto para implementação de NLP..... | 44 |
| Figura 55 - Método Turkey para identificação de outliers..... | 46 |
| Figura 56 - Instalação da biblioteca "freetype"..... | 47 |
| Figura 57 - Tratamento do texto com a função "limpa_texto"..... | 48 |
| Figura 58 - Gerando arquivo externo com texto "Descricao" processado | 48 |
| Figura 59 - Filtrando o dataset CAC..... | 49 |
| Figura 60 - Aplicação dos modelos BoW e TF-IDF..... | 49 |
| Figura 61 - Resultado do processamento do modelo BoW e TF-IDF | 50 |
| Figura 62 - Código Python para geração da nuvem de palavras..... | 50 |
| Figura 63 - Nuvem de palavras utilizando o formato do mapa do Brasil..... | 51 |
| Figura 64 - 30 palavras mais frequentes | 51 |
| Figura 65 - Gráfico distribuição das 30 palavras mais frequentes | 52 |
| Figura 66 - Representação da redução da dimensionalidade..... | 53 |
| Figura 67 - Tentativa de redução da dimensionalidade utilizando 3.000 componentes | 54 |
| Figura 68 - Análise do vector através da curva de cotovelo | 55 |
| Figura 69 - Cálculo e representação do coeficiente de Silhouette | 56 |
| Figura 70 - Treinamento e alocação em clusters | 57 |
| Figura 71 - Resultado da clusterização..... | 58 |
| Figura 72 - Palavras mais frequentes por cluster..... | 58 |
| Figura 73 - Preparação do dataset (cluster, palavra e TF-IDF) | 59 |
| Figura 74 - Plotagem dos 10 clusters | 59 |
| Figura 75 - Representação gráfica do TF-IDF do cluster e as 10 principais palavras | 60 |
| Figura 76 - Percentual de palavras por cluster..... | 60 |
| Figura 77 - Amostragem do cluster "0" | 61 |
| Figura 78 - Percentual de RVAT, Frota circulante e atendimento CAC por UF | 64 |
| Figura 79 – Mapa de calor: Concentração dos atendimentos via CAC | 65 |
| Figura 80 - Classificação do atendimento via CAC | 66 |

| | |
|---|----|
| Figura 81 - Alocação dos atendimentos por cluster..... | 66 |
| Figura 82 - 10 principais palavras constantes nos três maiores clusters | 67 |
| Figura 83 - Cálculo do TF-IDF..... | 67 |
| Figura 84 - Palavras com maior importância (TF-IDF) | 68 |
| Figura 85 - Análise da concordância da palavra "desbloqueio" | 69 |
| Figura 86 - Análise da similaridade da palavra "desbloqueio" | 69 |

1. Introdução

1.1. Contextualização

A compra e utilização de um produto ou contratação de um serviço passa por etapas que vão desde o primeiro contato com a marca até o pós-venda. Em linhas gerais, todos os trâmites deste processo fazem parte do que se convencionou a chamar de “jornada do cliente”, “jornada do consumidor” ou “jornada do usuário”. As três nomenclaturas se referem ao mapeamento de toda a experiência de compra e contratação de serviço.

Em todo o ciclo da “jornada do cliente” (touchpoints), a empresa fornecedora do produto ou serviço tem a oportunidade de receber dados e feedbacks do cliente/usuário. Com o avanço constante dos meios de comunicação e interação entre empresa e cliente/usuário, a entrada desses dados e feedbacks (inputs) está diversificada em diferentes plataformas: central de atendimento ao cliente (CAC), mídias sociais, plataformas de pesquisa direta e indireta, chatbots, aplicativos de mensagens, etc.

1.1.1 O que é a CAC

A central de atendimento ao cliente (CAC) é um departamento, área ou estrutura dentro da empresa onde são centralizadas as demandas dos clientes, ou seja, toda interação direta – relacionada a um serviço ou produto -, feita entre cliente e empresa é conduzida por esse canal.

A CAC utiliza os mais diversos meios de comunicação, desde um telefone de contato até um chatbot disponível no site da marca. Com o avanço tecnológico os clientes estão cada vez mais próximos das marcas, o que lhes possibilita interagir de forma mais fácil e com o fabricante/representante do produto e serviço (Figura 1 - Percentual de utilização dos canais de contato com o CAC).

| Canal de atendimento | % |
|----------------------|-----|
| Telefone | 65% |
| Emails | 15% |
| Mídias Sociais | 15% |
| Whatsapp | 5% |
| Chat | 1% |

Figura 1 - Percentual de utilização dos canais de contato com o CAC

Dessa forma, a CAC deixa de ser um lugar em que os profissionais de atendimento ficam reunidos atendendo ao telefone, e passa a ser um sistema que integra múltiplos canais.

De forma geral, os principais motivos do contato do Cliente a CAC são:

- Solicitar suporte;
- Tirar dúvidas;
- Compartilhar elogios e críticas;
- Fazer reclamações;
- Solicitar trocas;
- Buscar apoio para a utilização de um serviço ou produto contratado.

Segundo pesquisas, 93% das equipes de atendimento concordam que os clientes têm, atualmente, maiores expectativas em relação à experiência que terão com a marca, do que tinham há alguns anos. Os resultados que surgem como consequência de uma experiência negativa podem ser desastrosos para os objetivos da empresa.

Prioridades dos Clientes

- Em média, 56% dos clientes deixariam de comprar com uma empresa devido a uma má experiência de atendimento.
- 61 % dos consumidores afirmam que ser bem atendido é mais importante do que o preço ou a qualidade dos produtos.
- 72% esperam que os atendentes conheçam seu histórico com a empresa (como contatos, compras, suporte oferecido).
- Quase 30% dos clientes esperam resolver seus problemas via telefone em menos de cinco minutos.
- 7 em cada 10 consumidores dizem que pagaram mais caro para fazer negócios com uma empresa que oferece um ótimo serviço de atendimento.
- A geração do milênio (nascidas entre meados da década de 1985 e o fim dos anos 90) está disposta a gastar 21% a mais para receber um ótimo atendimento ao cliente.
- Metade dos clientes escolhe um canal com base na rapidez com que precisa de uma resposta.

1.1.2 O que é Customer Experience (CX)

Customer Experience (CX), ou Experiência do Cliente, em português, pode ser definido como o conjunto de impressões que o consumidor tem de uma empresa, a partir de todas as interações estabelecidas com ela ao longo da sua jornada de compra.

É muito comum confundir Customer Experience com atendimento ao cliente. A diferença é que o atendimento ao cliente se refere ao suporte prestado para esclarecimento de dúvidas e resolução de problemas.

Já o conceito de Customer Experience é mais amplo, envolvendo todos os pontos de interação entre o cliente e a marca, ou seja, desde o primeiro contato que o usuário tem com a empresa até o pós-venda.

Dessa forma, podemos dizer que o atendimento ao cliente é uma parte essencial do Customer Experience. O CX, porém, engloba muitos outros processos e setores necessários para proporcionar uma boa experiência para o consumidor e construir um relacionamento duradouro com ele.

Em um e-commerce, por exemplo, a experiência do cliente está diretamente relacionada a processos como o suporte ao consumidor, a estratégia de marketing e vendas do negócio, a interface de navegação do site, o processo de checkout e a entrega do pedido.

Dentro do CX, possuímos a ferramenta Net Promoter Score (NPS). Também. A Net Promoter Score é uma pesquisa que utiliza dados quantitativos e qualitativos para avaliar o quanto os clientes estão satisfeitos com a experiência que tiveram (ou estão tendo) com uma determinada empresa.

A Figura 2 - Perguntas feitas na pesquisa NPS detalha o questionamento feito durante a pesquisa NPS:

| Nr. Pergunta | Pergunta |
|--------------|--|
| P01 | Como você avalia sua experiência geral desde o 1º contato até o momento em que retirou o seu veículo da oficina? |
| P02 | Qual a possibilidade de voltar a utilizar os serviços desta concessionária? |
| P03 | Como você avalia o esforço da equipe da oficina em se relacionar com o cliente? |
| P04 | Como você avalia a disponibilidade de peças na concessionária para realização do serviço? |
| P05 | Finalização do veículo no prazo conforme prometido |
| P06 | Como você avalia a disponibilidade da oficina em realizar o serviço? |
| P07 | Como você avalia sua satisfação com a disponibilidade de seu veículo para voltar ao trabalho? |
| P08 | Como você avalia o preço de peças de reposição? |
| P09 | Como você avalia a qualidade de serviço e reparo? |
| P10 | Como você avalia os custos de reparo e manutenção? |

Figura 2 - Perguntas feitas na pesquisa NPS

A Figura 3 - Classificação do NPS apresenta a classificação do NPS de acordo com a nota final calculada na pesquisa.

| Pontuação | Classificação |
|-----------|---------------|
| 0 - 2 | Detratores |
| 3 | Neutros |
| 4 - 5 | Promotores |

Figura 3 - Classificação do NPS

1.2.3 Jornada do cliente

Na Figura 4 - Etapas da jornada do cliente na indústria automobilística, destacamos os 16 momentos (touch points) de contato do cliente com a marca (fabricante e rede de vendas e assistência técnica).

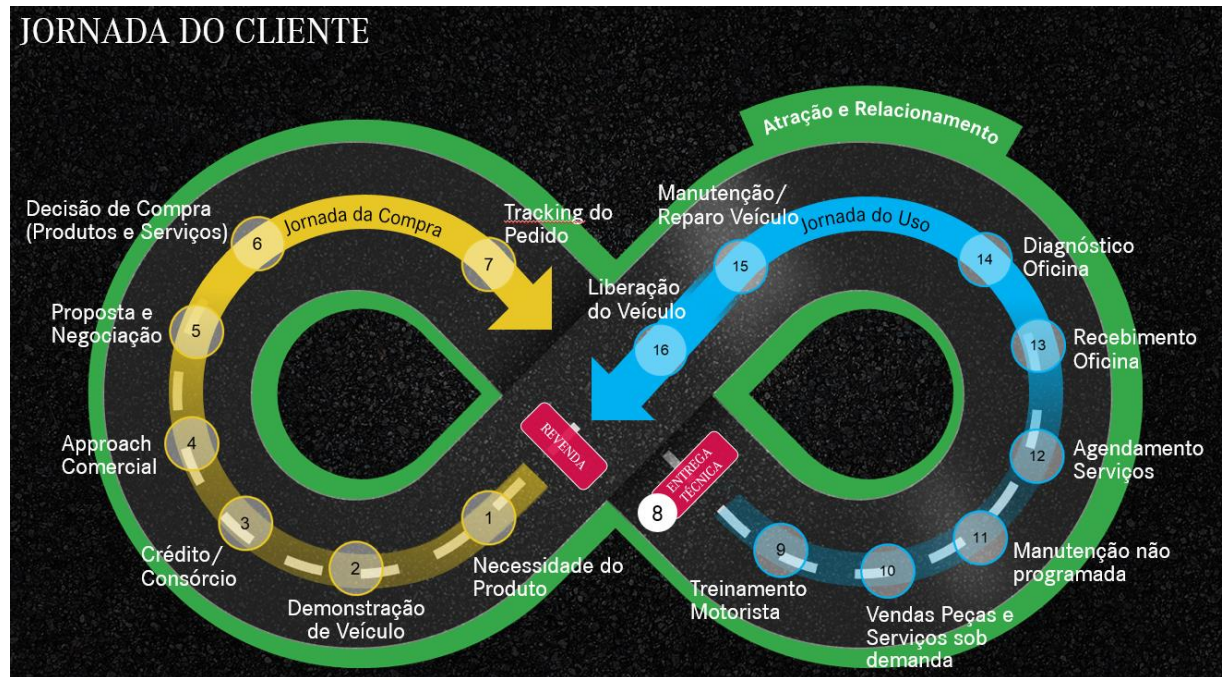


Figura 4 - Etapas da jornada do cliente na indústria automobilística

1.2. O problema proposto

Analisar os dados e feedback, sejam eles questionamentos, reclamações e elogios, recebidos dos clientes através da CAC de uma empresa automobilística, identificar potenciais fatores que possam contribuir para melhoria dos produtos, serviços e atendimento da própria CAC.

Através da metodologia 5W, descrevemos de forma objetiva o problema proposto (Figura 5 - 5W para auxílio para interpretação do problema proposto):

| W | Resposta |
|--|---|
| (Why?) Por que esse problema é importante? | As interações dos clientes com as empresas fornecedoras de produtos e serviços estão mais constantes e através de diversos canais no ciclo da "jornada do cliente". Esses dados/feedbacks (inputs) são valiosos para tomada de decisão para melhoria do produto, serviço e reversão de "detrações". |
| (Who?) De quem são os dados analisados? De um governo? Um ministério ou secretaria? Dados de clientes? | Dados e feedbacks recebidos de clientes de uma empresa automobilística (caminhões e ônibus) através da CAC (central de atendimento ao Cliente). |
| (What?): Quais os objetivos com essa análise? O que iremos analisar? | Analisaremos os dados e feedbacks (questionamentos, reclamações, elogios do detratores e promotores) para tomada de decisões de melhorias técnicas e comerciais do produto e serviço. |
| (Where?): Trata dos aspectos geográficos e logísticos de sua análise. | Nesse estudo, analisaremos os dados e feedbacks para os veículos caminhões e ônibus de um determinado fabricante que operam no território Brasileiro. |
| (When?): Qual o período está sendo analisado? A última semana? Os últimos 6 meses? O ano passado? | Analisaremos os dados/feedbacks dos anos de 2019, 2020, 2021, 2022 e 2023 (2023 até Jun) |

Figura 5 - 5W para auxílio para interpretação do problema proposto

1.3. Objetivos

Nessa dissertação, aplicaremos Processamento de Linguagem Natural (PLN, ou NLP em inglês) não supervisionado, treinando um modelo de análise de sentimentos dos contatos do cliente, recebidos através da CAC para os produtos e serviços da marca, desde a limpeza do dataset até a avaliação do modelo.

¹O processamento de linguagem natural é a área da ciência da computação focada na geração e compreensão das línguas humanas. Como a comunicação entre nós, humanos, fica restrita praticamente ao uso das línguas naturais, muitas são as tarefas que envolvem PLN. Hoje, já é possível detectar várias atividades do cotidiano das pessoas que envolvem essa área, seja através do uso de assistentes pessoais em celulares, filtros de spam na caixa de e-mails, tradutores automáticos e corretores ortográficos, por exemplo.

Os dados serão processados através dos modelos Bag of Words e TF-IDF e, agrupados em clusters pelo modelo K-Means de aprendizado de máquina não supervisionado. Utilizaremos a biblioteca scikit-learn, nltk e spaCy.

²K-Means é um algoritmo de clusterização (ou agrupamento) disponível na biblioteca Scikit-Learn. É um algoritmo de aprendizado não supervisionado (ou seja, que não precisa de inputs de confirmação externos) que avalia e clusteriza os dados de acordo com suas características, como por exemplo:

- 1) Lojas/centro logístico
- 2) Clientes/produtos ou serviços semelhantes
- 3) Clientes/características semelhantes
- 4) Séries/gênero da série ou faixa etária
- 5) Usuários de uma rede social/usuário influenciador
- 6) Paciente/sintoma ou característica semelhante

Nessa dissertação, as características dos dados concentram-se nos itens 1) e 2)

¹ <https://medium.com/turing-talks/introdu%C3%A7%C3%A3o-ao-processamento-de-linguagem-natural-com-baco-exu-do-blues-17cbb7404258>

² <https://medium.com/programadores-ajudando-programadores/k-means-o-que-%C3%A9-como-funciona-aplica%C3%A7%C3%B5es-e-exemplo-em-python-6021df6e2572>

2. Coleta de Dados

Os dados utilizados nesse estudo foram obtidos através do sistema de gerenciamento da CAC, cadastro da rede de vendas e assistência técnica (RVAT) de uma empresa automobilística.

Por questões relacionadas a proteção da marca e a lei LGPD, os dados foram alterados para impossibilitar a identificação do fabricante, cliente, rede de vendas e assistência técnica e o modelo dos veículos, sem quaisquer prejuízos ao estudo aqui proposto.

| # | Dataset | Formato |
|----|--|---------------------------------|
| 01 | <u>CAC.csv</u> : Base de dados de atendimento via CAC (Captação de dados através de ligação telefônica, e-mail e whatsapp e inseridos no sistema de gerenciamento da CAC) | .csv (original .xlsx) |
| 02 | <u>RVAT.csv</u> : Dados geográficos da rede de vendas e assistência técnica (somente para estatística geográfica) | .csv (original .accdb – Access) |
| 03 | <u>FrotaCirculante.xls</u> : Dados relativos à frota circulante por UF dos veículos Truck e Bus da empresa analisada | .xls |

Durante o desenvolvimento desse estudo, os datasets serão vinculados entre si para maximizar as análises.

Dataset: **CAC.csv**

| Nome da coluna/campo | Descrição | Tipo |
|----------------------|--|---|
| NrEvento | Código/número único que identifica o registro do contato feito pelo Cliente ao CAC. | String (Ex: 3-75192115167) |
| Classificacao | Identifica o motivo do contato do Cliente ao CAC | String (Ex.: Reclamação) |
| SubClassificacao | Um nível mais detalhado da Classificação. | String (Ex.: Qualidade Pós Venda) |
| Data | Data em que foi recebido o contato do Cliente ao CAC. | DateTime (Ex.: 07/06/2019 16:13:28) |
| Descricao | Descrição textual do motivo do contato do Cliente ao CAC. | String (Ex.: CLIENTE ENTROU EM CONTATO PARA SOLICITAR CODIGO DE RADIO) |
| ExpectativaCliente | O que o Cliente espera de resposta/solução para o questionamento feito à CAC. | String (Ex.: CLIENTE DIZ QUE PRECISA DE LOGIN E SENHA DO FLEETBOARD) |
| SegProdutoServico | Indica qual segmento de produto o Cliente está utilizando. Nesse estudo, trabalharemos com os segmentos “Truck” e “Bus”. | String (Truck ou Bus) |
| GSSN | Código/número único que identifica a Rede de Vendas e Assistência Técnica. Por esse campo chave, faremos o relacionamento com o dataset “RedeVendaAssistenciaTecnica.csv”. | String (Ex.: GS0003212) |

Dataset: ³**RVAT.csv**

| Nome da coluna/campo | Descrição | Tipo |
|----------------------|--|-----------------------------|
| GSSN | Código/número único que identifica a Rede de Vendas e Assistência técnica. Por esse campo chave, faremos o relacionamento com os dataset “CAC.csv” e “CX.csv”. | String (Ex.: GS0003212) |
| Municipio | Cidade na qual Rede de Vendas e Assistência Técnica está localizada. | String (Ex.: Contagem) |
| UF | O estado da federação onde o município está localizado. | String (Ex.: MG) |
| Latitude | Posição geográfica (latitude) da Rede de Vendas e Assistência Técnica. | String (Ex.: -19,961157) |
| Longitude | Posição geográfica (longitude) da Rede de Vendas e Assistência Técnica. | String (Ex.: -44,052236) |

Dataset: ⁴**FrotaCirculante.xlsx**

| Nome da coluna/campo | Descrição | Tipo |
|----------------------|--|---------------------------|
| UF | O estado da federação onde o município está localizado. | String (Ex.: MG) |
| FrotaCirculanteTotal | Posição geográfica (latitude) da Rede de Vendas e Assistência Técnica. | Int (Ex.: SP = 404459) |

³ Rede de venda e assistência técnica⁴ Referente a frota circulante do fabricante/marca estudada

3. Processamento/Tratamento de Dados

Para esse estudo, utilizaremos a linguagem de programação ⁵Python [3], processada no ambiente de desenvolvimento integrado “Jupyter Notebook 6.4.5” (parte da IDE ⁶anaconda), bibliotecas específicas do Python (comentadas no código fonte desse estudo), MS Excel (tratamento anônimo dos dados) e MS Access 2016.

3.1 Pré-processamento dos dados

3.1.1 Dataset CAC

3.1.1.1 Preparação dos dados do dataset CAC no formato .csv

Nessa etapa, fizemos o tratamento dos dados do dataset “CAC.xlsx” com a ferramenta “Excel” com a finalidade de descaracterizar os dados do fabricante, cliente, rede de vendas e assistência técnica e o modelo dos veículos. Esses dados foram substituídos por dados genéricos, mantendo a relação e plausibilidade do dataset.

Posteriormente, através da própria ferramenta “Excel”, o dataset foi exportado para o formato .csv (CAC.csv), usando o “;” como separador de campos e com identificação de string através das aspas (“”). Veja as Figura 6 - Descaracterização dos dados (sigilo e LGPD) e Figura 7 - Exportação dos dados para formato .csv:

| | A | B | C | D | E | F | G | H |
|----|---------------|---------------|----------------------------------|------------------|------------------------|--------------------------|-------------------|-----------|
| 1 | NrEvento | Classificacao | SubClassificacao | Data | Descricao | ExpectativaCliente | SegProdutoServico | GSSN |
| 2 | 3-75192115167 | Solicitação | Questionamento do produto | 26/07/2023 07:53 | -Nome do cliente: SR | 1e de contato: +; Truck | | |
| 3 | 3-75189947576 | Reclamação | Reclamação do produto | 26/07/2023 07:42 | SR ADENIR TELEFONE | IM O CHECKLIST Truck | | GS0021190 |
| 4 | 3-75166478203 | Solicitação | Questionamento ao Concessionário | 25/07/2023 19:42 | # ACESSO AO TCAS # | Truck | | GS0016478 |
| 5 | 3-75160845099 | Solicitação | Questionamento de vendas | 25/07/2023 18:51 | SRA SOLANGE TELEFONE | 12821 CPF/CNPJ Truck | | |
| 6 | 3-75159258086 | Solicitação | Questionamento de vendas | 25/07/2023 17:39 | SRA SELMA TELEFONE | ROTOCOLO 3-75 Truck | | |
| 7 | 3-75158847742 | Solicitação | Questionamento do Serviço | 25/07/2023 17:28 | SR EDSON TELEFONE | 58 SITUACAO – S Truck | | |
| 8 | 3-75158847668 | Reclamação | Qualidade Pós Venda | 25/07/2023 17:10 | SR MARCELO TELEFONE | IVA – SR MARCELO Truck | | GS0003262 |
| 9 | 3-75160968547 | Reclamação | Qualidade Pós Venda | 25/07/2023 17:05 | SR VINICIUS TELEFONE | JS DESEJA UMA F Truck | | GS0018104 |
| 10 | 3-75158847493 | Solicitação | Questionamento do Serviço | 25/07/2023 16:58 | SR ERSINIO TELEFONE | 313 CPF/CNPJ – (Truck | | |
| 11 | 3-75160895543 | Solicitação | Questionamento de vendas | 25/07/2023 16:57 | COMENTARIO=SOLICITACAO | ELO EMAIL CADASTRO Truck | | |
| 12 | 3-75159257818 | Reclamação | Qualidade Pós Venda | 25/07/2023 16:56 | SR ANDERSON TELEFONE | 1 QUIL QUANDO F Truck | | GS0003123 |

Figura 6 - Descaracterização dos dados (sigilo e LGPD)

⁵ Python é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte

⁶ O Anaconda IDE é uma distribuição open source da linguagem de programação Python..

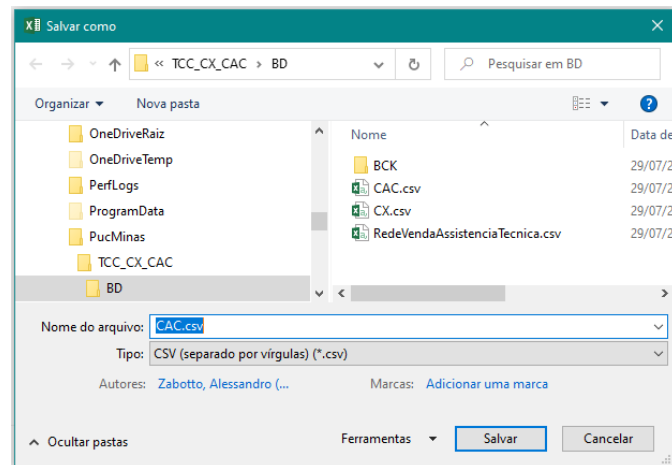


Figura 7 - Exportação dos dados para formato .csv

3.1.1.2 Leitura e análise do dataset CAC.csv

Com a arquivo CAC.csv pronto e alocado na pasta de trabalho (C:\PucMinas\TCC_CAC\BD), faremos a inicialização do Python através do Jupyter, definindo as bibliotecas e pasta padrão de trabalho.

- a) Faremos a instalação da biblioteca spaCy (biblioteca avançada para tratamento de NPL) através do prompt de comando do “anaconda”, com os dicionários em português baixados diretamente do site <https://spacy.io/models/pt>. Essa ação foi necessária pois o Jupyter acusava problemas na porta 443:

```
...>conda install -c conda-forge spacy
```

```
...>pip install C:\PucMinas\TCC_CAC\BD\pt_core_news_lg-3.6.0-py3-none-any.whl
```

```
...>pip install C:\PucMinas\TCC_CAC\BD\pt_core_news_md-3.6.0-py3-none-any.whl
```

```
...>pip install C:\PucMinas\TCC_CAC\BD\pt_core_news_sm-3.6.0-py3-none-any.whl
```

- b) Iniciaremos o código fonte através do Jupyter, instalando e importando as bibliotecas necessárias para a análise dos dados. Vamos desabilitar as mensagens de erros referentes a “DeprecationWarning” (Figura 8 - Instalação e importação das bibliotecas e Figura 9 - Desabilitando as mensagens de erros referentes a “DeprecationWarning”):

```

#Instalação das bibliotecas que utilizaremos
!pip install freetype-py
!pip install -U pip setuptools wheel
!pip install -U spacy
!python -m spacy download pt_core_news_sm
!pip install unicode
!pip install wordcloud
!pip install folium
!pip install yellowbrick
!pip install opencv-python

#Principais bibliotecas Python para trabalhar com NLP e plotagem de gráficos
import pandas as pd          #manipulação e análise de dados
import numpy as np           #trabalhos matemáticos
import re                    #tratamento de expressões regulares
import nltk                  #tratamento de linguagem natural (PLN ou NPL)
import spacy                 #mesmo conceito do nltk mais com recursos mais avançados
import matplotlib.pyplot as plt #plotagem gráficos 2D/3D, visualizações estáticas, animadas e interativas
import seaborn as sns        #criação de gráficos estatísticos elegantes e informativos
import os                    #comandos do sistema operacional
get_ipython().run_line_magic('matplotlib', 'inline')

```

Figura 8 - Instalação e importação das bibliotecas

```

#código para inibir as mensagens do sistema (Ex.: DeprecationWarning)
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")

```

Figura 9 - Desabilitando as mensagens de erros referentes a “DeprecationWarning”

- c) Para facilitar a leitura dos datasets, definimos a pasta padrão de trabalho através da biblioteca “os”:

```

os.chdir('C:\PucMinas\TCC_CAC_CX\BD') #definição do diretório padrão onde constam os datasets

```

- d) Nosso próximo passo é a leitura do dataset “CAC.csv”, utilizando “;” como separador e encoding “cp1252” e verificando as informações após a importação (Figura 10 - Leitura do dataset CAC):

```
#Leitura do dataset CAC
dsCAC = pd.read_csv('CAC.csv', sep=";", encoding = 'cp1252')

#informações gerais sobre o dataset dsCAC
#Ex.: qtde de registros, colunas, tipo dos dados, etc.
dsCAC.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73029 entries, 0 to 73028
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   NrEvento               73029 non-null  object
1   Classificacao          73029 non-null  object
2   SubClassificacao       73029 non-null  object
3   Data                   73029 non-null  object
4   Descricao              72877 non-null  object
5   ExpectativaCliente     19025 non-null  object
6   SegProdutoServico      73029 non-null  object
7   GSSN                   24894 non-null  object
dtypes: object(8)
memory usage: 4.5+ MB
```

Figura 10 - Leitura do dataset CAC

Podemos verificar que a coluna “GSSN” possui 48.134 registros nulos (sem dados). Em determinadas análises, utilizaremos essa coluna para se relacionar com o dataset RVAT, sendo necessário desconsiderarmos os registros nulos.

- e) Vamos analisar os 5 primeiros e 5 últimos registros do dataset CAC e fazermos as primeiras intervenções no dataset (Figura 11 - Leitura dos 5 primeiros registros do dataset CAC e Figura 12 - Leitura dos 5 últimos registros do dataset CAC):

dsCAC.head()

| | NrEvento | Classificacao | SubClassificacao | Data | Descricao | ExpectativaCliente | SegProdutoServico | GSSN |
|---|---------------|---------------|----------------------------------|--------------------|---|---|-------------------|-----------|
| 0 | 3-75192115167 | Solicitação | Questionamento do produto | 26/7/2023 00:00:00 | : BRUNO de contato: INFORMAÇÃO: CLIENTE ... | NaN | Truck | NaN |
| 1 | 3-75189947576 | Reclamação | Reclamação do produto | 26/7/2023 00:00:00 | ADENIR CHASSI BMNB CONCESSIONARIO ENVOLVIDO ... | CLIENTE TEM O CHECKLIST E DESEJA QUE SEJA COMP... | Truck | GS0021190 |
| 2 | 3-75166478203 | Solicitação | Questionamento ao Concessionário | 25/7/2023 00:00:00 | # ACESSO AO TCAS # | NaN | Truck | GS0016478 |
| 3 | 3-75160845099 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SOLANGE / - SITUACAO - A SOLANGE ENTROU... | NaN | Truck | NaN |
| 4 | 3-75159258086 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SELMA PROTOCOLO A SELMA ENTROU EM CONTA... | NaN | Truck | NaN |

Figura 11 - Leitura dos 5 primeiros registros do dataset CAC

dsCAC.tail()

| | NrEvento | Classificacao | SubClassificacao | Data | Descricao | ExpectativaCliente | SegProdutoServico | GSSN |
|-------|--------------|---------------|----------------------------------|--------------------|---|-------------------------------|-------------------|-----------|
| 73024 | 2-1316718352 | Solicitação | Questionamento do produto | 10/4/2019 00:00:00 | Informa que no painel apareceu a mensagem nive... | NaN | Truck | NaN |
| 73025 | 2-1317883767 | Solicitação | Questionamento do Serviço | 10/4/2019 00:00:00 | R Nepomuceno questiona sobre ativação do serviço. | NaN | Truck | NaN |
| 73026 | 2-1317864044 | Reclamação | Reclamação do produto | 10/4/2019 00:00:00 | . Seler informa que o veículo está na rede com... | Solicita uuma nova avaliação. | Truck | GS0003205 |
| 73027 | 2-1316802925 | Solicitação | Questionamento ao Concessionário | 10/4/2019 00:00:00 | PROCESSO DE PAGAMENTO. Boa tarde, SG: | NaN | Truck | GS0025690 |
| 73028 | 2-1316717916 | Solicitação | Questionamento do produto | 10/4/2019 00:00:00 | Informa que o rádio esta bloqueado deseja auxi... | NaN | Truck | NaN |

Figura 12 - Leitura dos 5 últimos registros do dataset CAC

Podemos fazer nossa primeira manipulação do dataset CAC, eliminando os registros duplicados, eliminando a coluna NrEvento e concatenando as colunas “Descricao” e “ExpectativaCliente” pois ambas são texto captados durante o contato do cliente e queremos analisá-las conjuntamente:

Verificando registros duplicados (não existem registros duplicados, Figura 13 - Análise de registros duplicados):

```
#verificando quantos registros temos duplicados no dataset
dsCAC.duplicated().value_counts()

False    73029
dtype: int64
```

Figura 13 - Análise de registros duplicados

Excluindo a coluna “NrEvento” (Figura 14 - Exclusão da coluna "NrEvento"):

```
#exclusão da coluna NrEvento
dsCAC.drop(['NrEvento'],axis=1,inplace=True)
```

Figura 14 - Exclusão da coluna "NrEvento"

Antes de fazermos a concatenação, vamos criar duas colunas adicionais contendo o tamanho em caracteres da coluna “Descricao” e “ExpectativaCliente” para podermos comparar o tamanho após a concatenação (Figura 15 - Calculando o tamanho das colunas "Descricao" e "ExpectativaCliente"):

```
#incluindo a coluna TamDescricao e TamExpectativaCliente com a qtd de caracteres contidos
#em cada coluna
dsCAC['Descricao'].fillna('',inplace=True) #preenchendo os registros nulos
dsCAC['ExpectativaCliente'].fillna('',inplace=True) #preenchendo os registros nulos
dsCAC.loc[:, 'TamDescricao'] = dsCAC.Descricao.apply(lambda x: len(str(x)))
dsCAC.loc[:, 'TamExpectativaCliente'] = dsCAC.ExpectativaCliente.apply(lambda x: len(str(x)))

dsCAC.head()
```

| | Classificacao | SubClassificacao | Data | Descricao | ExpectativaCliente | SegProdutoServico | GSSN | TamDescricao | TamExpectativaCliente |
|---|---------------|----------------------------------|--------------------|---|---|-------------------|-----------|--------------|-----------------------|
| 0 | Solicitação | Questionamento do produto | 26/7/2023 00:00:00 | : BRUNO de contato: INFORMAÇÃO: CLIENTE ... | | Truck | NaN | 100 | 0 |
| 1 | Reclamação | Reclamação do produto | 26/7/2023 00:00:00 | ADENIR CHASSI BMNB CONCESSIONARIO ENVOLVIDO ... | CLIENTE TEM O CHECKLIST E DESEJA QUE SEJA COMP... | Truck | GS0021190 | 427 | 53 |
| 2 | Solicitação | Questionamento ao Concessionário | 25/7/2023 00:00:00 | # ACESSO AO TCAS # | | Truck | GS0016478 | 18 | 0 |
| 3 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SOLANGE / - SITUACAO - A SOLANGE ENTROU... | | Truck | NaN | 340 | 0 |
| 4 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SELMA PROTOCOLO A SELMA ENTROU EM CONTA... | | Truck | NaN | 93 | 0 |

Figura 15 - Calculando o tamanho das colunas "Descricao" e "ExpectativaCliente"

Agora vamos concatenar as colunas e verificar se o tamanho em caracteres se alterou (Figura 16 - Concatenando as colunas "Descricao" e "ExpectativaCliente"):

```
#concatenado (juntando) as colunas Descricao e ExpectativaCliente
dsCAC['ExpectativaCliente'].fillna('',inplace=True) #preenchendo os registros nulos da coluna ExpectativaCliente
dsCAC['Descricao'] = (dsCAC['Descricao'] + " " + dsCAC['ExpectativaCliente']) #fazendo a concatenação
dsCAC['Descricao'] = dsCAC.Descricao.apply(lambda x: "" if str(x) == "" else x)

#atualizando a coluna TamanhoDescricao com a qtd de caracteres contidos na coluna Descricao (após concatenação)
dsCAC.loc[:, 'TamDescricao'] = dsCAC.Descricao.apply(lambda x: len(str(x)))
dsCAC.head()
```

| | Classificacao | SubClassificacao | Data | Descricao | ExpectativaCliente | SegProdutoServico | GSSN | TamDescricao | TamExpectativaCliente |
|---|---------------|----------------------------------|--------------------|---|---|-------------------|-----------|--------------|-----------------------|
| 0 | Solicitação | Questionamento do produto | 26/7/2023 00:00:00 | : BRUNO de contato: INFORMAÇÃO: CLIENTE ... | | Truck | NaN | 101 | 0 |
| 1 | Reclamação | Reclamação do produto | 26/7/2023 00:00:00 | ADENIR CHASSI BMNB CONCESSIONARIO ENVOLVIDO ... | CLIENTE TEM O CHECKLIST E DESEJA QUE SEJA COMP... | Truck | GS0021190 | 481 | 53 |
| 2 | Solicitação | Questionamento ao Concessionário | 25/7/2023 00:00:00 | # ACESSO AO TCAS # | | Truck | GS0016478 | 19 | 0 |
| 3 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SOLANGE / - SITUAÇÃO - A SOLANGE ENTROU... | | Truck | NaN | 341 | 0 |
| 4 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SELMA PROTOCOLO A SELMA ENTROU EM CONTA... | | Truck | NaN | 94 | 0 |

Figura 16 - Concatenando as colunas "Descricao" e "ExpectativaCliente"

Podemos comprovar através do registro 1 que a quantidade de caracteres da coluna “Descricao” alterou de 427 para 481. Dessa forma, podemos excluir as colunas “ExpectativaCliente” e “TamExpectativaCliente”. Vamos manter a coluna “TamDescricao” para ser utilizada posteriormente (Figura 17 - Exclusão das colunas "ExpectativaCliente" e "TamExpectativaCliente" e Figura 18 - Análise dos 5 primeiros registros após exclusão das colunas):

```
#excluindo a coluna ExpectativaCliente e TamExpectativaCliente
dsCAC.drop(['ExpectativaCliente'],axis=1,inplace=True)
dsCAC.drop(['TamExpectativaCliente'],axis=1,inplace=True)
```

Figura 17 - Exclusão das colunas "ExpectativaCliente" e "TamExpectativaCliente"

```
#mostrando os 5 primeiros registros
dsCAC.head()
```

| | Classificacao | SubClassificacao | Data | Descricao | SegProdutoServico | GSSN | TamDescricao |
|---|---------------|----------------------------------|--------------------|---|-------------------|-----------|--------------|
| 0 | Solicitação | Questionamento do produto | 26/7/2023 00:00:00 | : BRUNO de contato: INFORMAÇÃO: CLIENTE ... | Truck | NaN | 101 |
| 1 | Reclamação | Reclamação do produto | 26/7/2023 00:00:00 | ADENIR CHASSI BMNB CONCESSIONARIO ENVOLVIDO ... | Truck | GS0021190 | 481 |
| 2 | Solicitação | Questionamento ao Concessionário | 25/7/2023 00:00:00 | # ACESSO AO TCAS # | Truck | GS0016478 | 19 |
| 3 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SOLANGE / - SITUAÇÃO - A SOLANGE ENTROU... | Truck | NaN | 341 |
| 4 | Solicitação | Questionamento de vendas | 25/7/2023 00:00:00 | A SELMA PROTOCOLO A SELMA ENTROU EM CONTA... | Truck | NaN | 94 |

Figura 18 - Análise dos 5 primeiros registros após exclusão das colunas

Na leitura do dataset CAC, o campo “Data” foi reconhecido como sendo “object”. Vamos convertê-lo para o formato de “datetime” para podermos explorar a análise dos dados de forma mais precisa. Antes, vamos aplicar um tratamento para acertar o formato da data de “d/m/yyyy” (tipo Object) para “yyyy/m/d” mantendo a coluna como “Object” (Figura 19 - Tratamento da coluna "Data"):

Nota: O desenvolvimento da função “AcertaData” foi necessário pois após várias tentativas, não foi possível fazer a conversão direta do formato da data de “d/m/yyyy” (tipo Object) para “yyyy/m/d” (tipo Object).

```
#função para inverter a data do formato "d/m/yyyy" para "yyy/m/d"
def AcertaData(strData):
    strDataSplit = strData.replace(" ", "00:00:00", "")
    strDataSplit = strDataSplit.split("/")
    Ano = str(strDataSplit[2])
    Mes = str("0" + strDataSplit[1])
    Mes = Mes[-2:]
    Dia = str("0" + strDataSplit[0])
    Dia = Dia[-2:]
    strDataSplit = Ano + "/" + Mes + "/" + Dia
    return strDataSplit
```

```
#Alterar ordem do campo data para "aaaa/mm/dd"
dsCAC['Data'] = dsCAC.Data.apply(AcertaData)
```

```
#mostrando 5 registro aleatórios
dsCAC.sample(5)
```

| | Classificacao | SubClassificacao | Data | Descricao | SegProdutoServico | GSSN | TamDescricao |
|-------|---------------|----------------------------------|------------|---|-------------------|-----------|--------------|
| 52320 | Reclamação | Qualidade Pós Venda | 2020/03/04 | . JOSE ENTRA EM CONTATO E INFORMA QUE EFETUOU ... | Truck | GS0003130 | 278 |
| 24647 | Solicitação | Questionamento do Serviço | 2021/10/18 | . MARCOS : CLIENTE GOSTARIA DO CONTATO DE CO... | Truck | NaN | 60 |
| 19093 | Solicitação | Questionamento do Serviço | 2022/01/12 | . MATEUS : / : CONCESSIONÁRIO ENVOLVIDO: ... | Truck | NaN | 205 |
| 306 | Solicitação | Questionamento ao Concessionário | 2023/06/02 | GS Boa Tarde O problema relatado abaixo ... | Truck | GS0003206 | 1060 |
| 29955 | Solicitação | Questionamento ao Concessionário | 2021/06/09 | fechamento | Truck | GS0003100 | 11 |

Figura 19 - Tratamento da coluna "Data"

Agora faremos a conversão da coluna “Data” para o formato “DateTime” (Figura 20 - Conversão do tipo da coluna "Data"):

```
#converter o campo "Data" para formato datetime
dsCAC['Data'] = pd.to_datetime(dsCAC['Data'])
```

```
dsCAC.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73029 entries, 0 to 73028
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Classificacao          73029 non-null  object
1   SubClassificacao       73029 non-null  object
2   Data                   73029 non-null  datetime64[ns]
3   Descricao              73029 non-null  object
4   SegProdutoServico      73029 non-null  object
5   GSSN                   24894 non-null  object
6   TamDescricao           73029 non-null  int64
dtypes: datetime64[ns](1), int64(1), object(5)
memory usage: 3.9+ MB
```

Figura 20 - Conversão do tipo da coluna "Data" para datetime

Para fazermos algumas análises nos dados, vamos incluir duas colunas no dataset CAC, chamadas de “Mes” e “Ano”, extraídas da coluna “Data” (Figura 21 - Inclusão das colunas "Mes" e "Ano" e Figura 22 - Leitura dos 5 primeiros registros do dataset CAC com as colunas mês e ano):

```
#para análise estatística, vamos incluir duas colunas,
#extraídas do campo "Data": "Mes" e "Ano"
dsCAC.loc[:, 'Mes'] = dsCAC['Data'].dt.month
dsCAC.loc[:, 'Ano'] = dsCAC['Data'].dt.year
dsCAC.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73029 entries, 0 to 73028
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Classificacao          73029 non-null object
1   SubClassificacao       73029 non-null object
2   Data                   73029 non-null datetime64[ns]
3   Descricao              73029 non-null object
4   SegProdutoServico      73029 non-null object
5   GSSN                   24894 non-null object
6   TamDescricao           73029 non-null int64
7   Mes                    73029 non-null int64
8   Ano                    73029 non-null int64
dtypes: datetime64[ns](1), int64(3), object(5)
memory usage: 5.0+ MB
```

Figura 21 - Inclusão das colunas "Mes" e "Ano"

dsCAC.head()

| | Classificacao | SubClassificacao | Data | Descricao | SegProdutoServico | GSSN | TamDescricao | Mes | Ano |
|---|---------------|----------------------------------|------------|---|-------------------|-----------|--------------|-----|------|
| 0 | Solicitação | Questionamento do produto | 2023-07-26 | : BRUNO de contato: INFORMAÇÃO: CLIENTE ... | Truck | NaN | 101 | 7 | 2023 |
| 1 | Reclamação | Reclamação do produto | 2023-07-26 | ADENIR CHASSI BMNB CONCESSIONARIO ENVOLVIDO ... | Truck | GS0021190 | 481 | 7 | 2023 |
| 2 | Solicitação | Questionamento ao Concessionário | 2023-07-25 | # ACESSO AO TCAS # | Truck | GS0016478 | 19 | 7 | 2023 |
| 3 | Solicitação | Questionamento de vendas | 2023-07-25 | A SOLANGE / - SITUACAO - A SOLANGE ENTROU... | Truck | NaN | 341 | 7 | 2023 |
| 4 | Solicitação | Questionamento de vendas | 2023-07-25 | A SELMA PROTOCOLO A SELMA ENTROU EM CONTA... | Truck | NaN | 94 | 7 | 2023 |

Figura 22 - Leitura dos 5 primeiros registros do dataset CAC com as colunas mês e ano

Com essa etapa, finalizamos o tratamento dos dados do CAC.

3.1.2 Dataset RVAT

3.1.2.1 Preparação dos dados do dataset RVAT no formato .csv

Nessa etapa, faremos o tratamento dos dados do dataset “RVAT.xlsx” com a ferramenta “Excel” e exportamos para o formato “.csv”, utilizando o “;” como separador de campos e com identificação de string através das aspas (“”). Esse dataset contém informações geográficas e será utilizado para fazer o relacionamento com o dataset CAC (Figura 23 - Tratamento dos dados do dataset “RVAT.xlsx”):

| | A | B | C | D | E |
|---|------------|---------------|----|------------|------------|
| 1 | GSSN | Município | UF | Latitude | Longitude |
| 2 | GS0003164 | PORTO ALEGRE | RS | -29,973875 | -51,17692 |
| 3 | GS0043890 | PELOTAS | RS | -31,732629 | -52,389668 |
| 4 | GS0048000 | NOVO HAMBURGO | RS | -29,659566 | -51,138767 |
| 5 | GS00105202 | SÃO PAULO | SP | -23,517441 | -46,579507 |
| 6 | GS0016459 | CAMPINAS | SP | -22,847143 | -47,091476 |
| 7 | GS0049230 | ITUMBIARA | GO | -18,366497 | -49,21286 |

Figura 23 - Tratamento dos dados do dataset “RVAT.xlsx”

3.1.2.2 Leitura e análise do dataset RVAT.csv

Após a exportação do dataset para o formato “.csv” através do Excel e alocação do arquivo na pasta de trabalho, faremos a leitura através do módulo pandas do Python e as análises gerais dos dados.

Nessa primeira análise, podemos verificar que é um dataset simples, com poucos registros e com nenhum campo nulo (Figura 24 - leitura do dataset RVAT):

```
#Leitura do dataset RVAT (rede de vendas e assistência técnica)
dsRVAT = pd.read_csv('RVAT.csv', sep=";", encoding = 'cp1252', decimal=',')

#informações gerais sobre o dataset dsRVAT
#Ex.: qtde de registros, colunas, tipo dos dados, etc.
dsRVAT.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221 entries, 0 to 220
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   GSSN        221 non-null    object
 1   Município   221 non-null    object
 2   UF          221 non-null    object
 3   Latitude    221 non-null    float64
 4   Longitude   221 non-null    float64
dtypes: float64(2), object(3)
memory usage: 8.8+ KB
```

Figura 24 - leitura do dataset RVAT

Utilizando o método “sample”, podemos visualizar uma amostragem aleatória dos dados do dataset, podendo definir quantos registros queremos na amostra (Figura 25 - Amostragem dos registros do dataset RVAT):

```
#amostragem de dados utilizando o "sample"
dsRVAT.sample(6)
```

| | GSSN | Município | UF | Latitude | Longitude |
|-----|------------|-------------------------|----|------------|------------|
| 72 | GS0016492 | UBERABA | MG | -19.777658 | -47.929198 |
| 139 | GS0043797 | CAMBÉ | PR | -23.245600 | -51.250892 |
| 0 | GS0003164 | PORTO ALEGRE | RS | -29.973875 | -51.176920 |
| 143 | GS00110503 | MARECHAL CÂNDIDO RONDON | PR | -24.543360 | -54.044781 |
| 102 | GS00106242 | ENTRE-IJUIS | RS | -28.378550 | -54.262920 |
| 126 | GS0003188 | ITABUNA | BA | -14.807160 | -39.296500 |

Figura 25 - Amostragem dos registros do dataset RVAT

Outra leitura que faremos é ordenar o dataset pelo campo “UF” e posteriormente fazermos o uso do método “unique” para retornar somente uma vez cada registro do campo “UF” (Figura 26 - Visualizando todas as UF que constam no dataset RVAT):

```
#ordenando o data set e retornando as UF
dsRVAT.sort_values('UF', ascending=True, inplace=True)
print(dsRVAT['UF'].unique())

['AC' 'AL' 'AM' 'AP' 'BA' 'CE' 'DF' 'ES' 'GO' 'MA' 'MG' 'MS' 'MT' 'PA'
 'PB' 'PE' 'PI' 'PR' 'RJ' 'RN' 'RO' 'RR' 'RS' 'SC' 'SE' 'SP' 'TO']
```

Figura 26 - Visualizando todas as UF que constam no dataset RVAT

Concluimos o tratamento do dataset RVAT.

4. Análise e Exploração dos Dados

4.1 – Dataset CAC

O dataset CAC é composto pelos registros de atendimentos ao Cliente de forma receptiva, isto é, é iniciativa do Cliente o contato com a CAC com a finalidade de solicitar suporte, orientação, fazer algum questionamento comercial e/ou técnico, expressar alguma insatisfação, reclamação, elogio, solicitar suporte de guincho, etc. Após a abertura do protocolo de atendimento, o operador/a da CAC utiliza o mesmo protocolo e registro para acompanhar a solicitação do Cliente até o seu encerramento.

Nesse dataset CAC, além da coluna com o principal dado (“Descricao”) que iremos explorar nessa dissertação através da metodologia NLP, possuímos outros dados que são captados durante o atendimento que nos fornecem ótimas opções para exploração dos dados e entendimento do perfil dos contatos recebidos pelos Clientes.

- a) A coluna “SegProdutoServico” identifica qual unidade de negócios o atendimento pertence. Nosso objetivo é analisarmos somente os atendimentos para as unidades de negócio “Truck” ou “Bus”. Vamos verificar se existe alguma alocação feita para unidade de negócio fora do escopo do estudo (Figura 27 - Segmentos que constam no dataset CAC):

```
#Verificando os segmentos do produto e serviço
dsCAC['SegProdutoServico'].value_counts()
```

| | |
|-------------------------|-------|
| Truck | 68252 |
| Não relacionado a marca | 2492 |
| Bus | 2158 |
| Car | 89 |
| Van | 38 |

Name: SegProdutoServico, dtype: int64

Figura 27 - Segmentos que constam no dataset CAC

Encontramos 2.616 registros que não fazem parte do nosso estudo. Vamos atualizar nosso dataset CAC, filtrando somente as unidades de negócio “Truck” ou “Bus” (Figura 28 - Dataset CAC após aplicar o filtro por "Truck" e "Bus"):

```
#Mantendo no dataset dsCAC somente os registros que são das unidades de negócio Truck ou Bus
dsCAC = dsCAC.query('SegProdutoServico=="Truck" or SegProdutoServico=="Bus"')

dsCAC.info()
dsCAC['SegProdutoServico'].value_counts()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 70410 entries, 0 to 73028
Data columns (total 9 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Classificacao        70410 non-null  object
1   SubClassificacao     70410 non-null  object
2   Data                 70410 non-null  datetime64[ns]
3   Descricao            70410 non-null  object
4   SegProdutoServico    70410 non-null  object
5   GSSN                 24515 non-null  object
6   TamDescricao         70410 non-null  int64
7   Mes                  70410 non-null  int64
8   Ano                  70410 non-null  int64
dtypes: datetime64[ns](1), int64(3), object(5)
memory usage: 5.4+ MB

Truck    68252
Bus       2158
Name: SegProdutoServico, dtype: int64
```

Figura 28 - Dataset CAC após aplicar o filtro por "Truck" e "Bus"

Originalmente, nossa dataset CAC possuía 73.028 registros e agora passamos a ter 70.410 registros.

Os atendimentos para a unidade de negócios “Bus” correspondem somente a 3,0% de todos os atendimentos feitos pela CAC. Essa diferença é reflexo da diferença na operação e manutenção dos veículos, onde para “Bus” os clientes possuem (em sua maioria) oficinas próprias operando 24x7x365, não acionando a CAC com frequência.

- b) Analisando o volume de atendimento por ano, verificamos que no ano de 2020 ocorreu um pico. Esse alto volume é resultado do fechamento temporário e/ou redução do atendimento na rede de vendas e assistência técnica, levando os Clientes a contatarem a CAC para saberem como deveriam agir para poderem reparar os veículos durante o lock-down (Figura 29 – Código p/ plotagem do gráfico comparativo atendimentos CAC / Ano e Figura 30 - Gráfico comparativo de atendimento CAC / Ano):

```
#Demonstração gráfica do volume de atendimento por ano
fig, ax = plt.subplots(figsize=(8,5))
#sns.countplot(dsCAC['Ano'],palette='YlGnBu')
sns.countplot(dsCAC['Ano'],palette='rocket')
plt.title('Atendimentos CAC por ano',fontsize=16, fontweight='bold')
ax.set_xlabel('Ano',fontsize=16, fontweight='bold')
ax.set_ylabel('Qtde. atendimentos',fontsize=12, fontweight='bold')
plt.show()
```

Figura 29 – Código p/ plotagem do gráfico comparativo atendimentos CAC / Ano

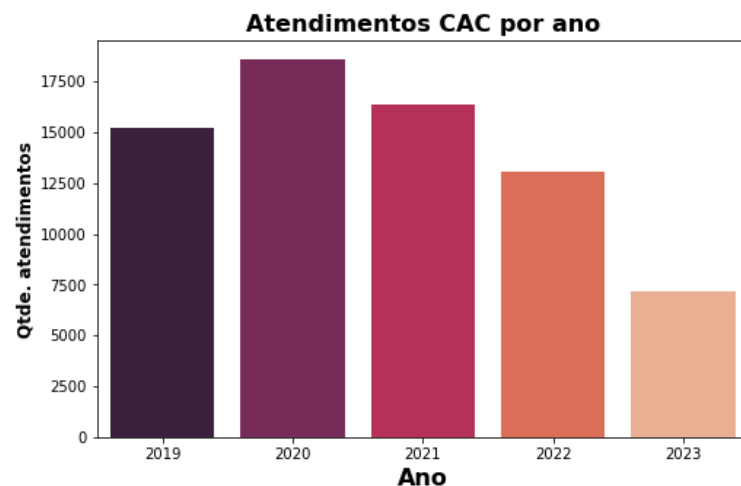


Figura 30 - Gráfico comparativo de atendimento CAC / Ano

Observamos também que em 2019 não possuímos os registros de atendimentos do ano completo (início: Abr/2019) e em 2023, os dados foram extraídos em Jul/2023 (Figura 31 - Primeira e última data de atendimento CAC constante no dataset CAC):

```
#primeiro e último Ano/Mês dos dados do dataset
print(dsCAC['Data'].min())
print(dsCAC['Data'].max())
```

2019-04-10 00:00:00
2023-07-26 00:00:00

Figura 31 - Primeira e última data de atendimento CAC constante no dataset CAC

- c) Uma outra forma de analisarmos os atendimentos, é explorando os dados da coluna “Classificacao”. No momento do atendimento via CAC, o atendente categoriza o atendimento utilizando algumas categorias pré-definidas (Figura 32 - Visualização dos dados por classificação do atendimento CAC):

```

dsCAC_Agrup = dsCAC_Agrup.sort_values(by=['Classificacao'],ascending=True)
dsCAC_Agrup = dsCAC_Agrup.groupby(["Classificacao"])
dsCAC_Agrup.describe()

```

| | TamDescricao | | | | Mes | | | | | | | | Ano | | | | | | | |
|---------------|--------------|------------|------------|-----|--------|-------|-------|--------|---------|----------|-----|-----|------|---------|-------------|----------|--------|--|--|--|
| | count | mean | std | min | 25% | 50% | 75% | max | count | mean | ... | 75% | max | count | mean | std | min | | | |
| Classificacao | | | | | | | | | | | | | | | | | | | | |
| Agendamento | 2058.0 | 280.006317 | 149.765756 | 2.0 | 200.25 | 255.0 | 339.0 | 1197.0 | 2058.0 | 6.224004 | ... | 8.0 | 12.0 | 2058.0 | 2019.714286 | 0.828515 | 2019.0 | | | |
| Reclamação | 16414.0 | 483.838857 | 352.434541 | 3.0 | 253.00 | 370.0 | 663.0 | 2203.0 | 16414.0 | 6.409285 | ... | 9.0 | 12.0 | 16414.0 | 2020.935543 | 1.262380 | 2019.0 | | | |
| Solicitação | 51938.0 | 166.417382 | 194.026818 | 1.0 | 52.00 | 116.0 | 223.0 | 1997.0 | 51938.0 | 6.464812 | ... | 9.0 | 12.0 | 51938.0 | 2020.657130 | 1.272284 | 2019.0 | | | |

Figura 32 - Visualização dos dados por classificação do atendimento CAC

Podemos também visualizar de forma gráfica os atendimentos agrupados pelo ano e classificação. Para isso, percorremos o dataset CAC, resumizando os dados em listas e dicionário para fazermos a plotagem do gráfico em tempo de execução (Figura 33 - Gráfico dos atendimentos CAC por ano e classificação):

```
#Visualizando em forma gráfica os atendimentos por ano e classificação
```

```
fig, ax = plt.subplots(figsize=(8,5))
sns.countplot(data=dsCAC, x="Ano", hue="Classificacao")
plt.grid(True, axis='y')
plt.legend(loc = "upper right")
plt.show()
```

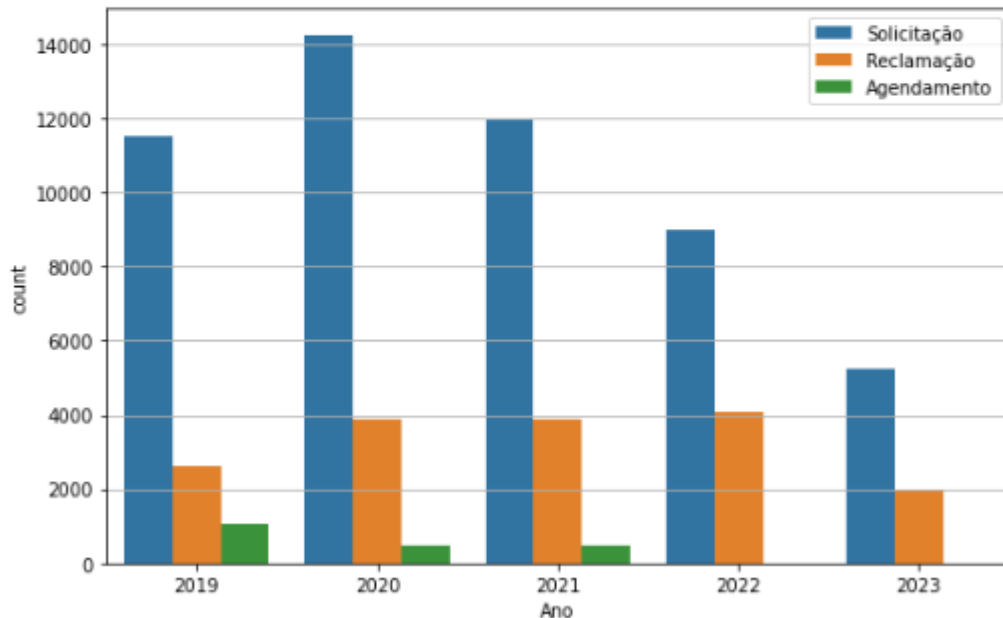


Figura 33 - Gráfico dos atendimentos CAC por ano e classificação

Notamos nos dados acima que praticamente não existem mais atendimentos relacionados a agendamento, reflexo da melhoria dos serviços prestados pela rede de vendas e assistência técnica (estrutura, qualificação e uso da tecnologia para gerenciamento dos agendamentos nas oficinas).

- d) Analisando o tamanho do texto da coluna “Descricao”, podemos identificar que o tamanho da descrição do atendimento concentra-se até 400 caracteres (Figura 34 – Distribuição e concentração do “TamDescricao”):

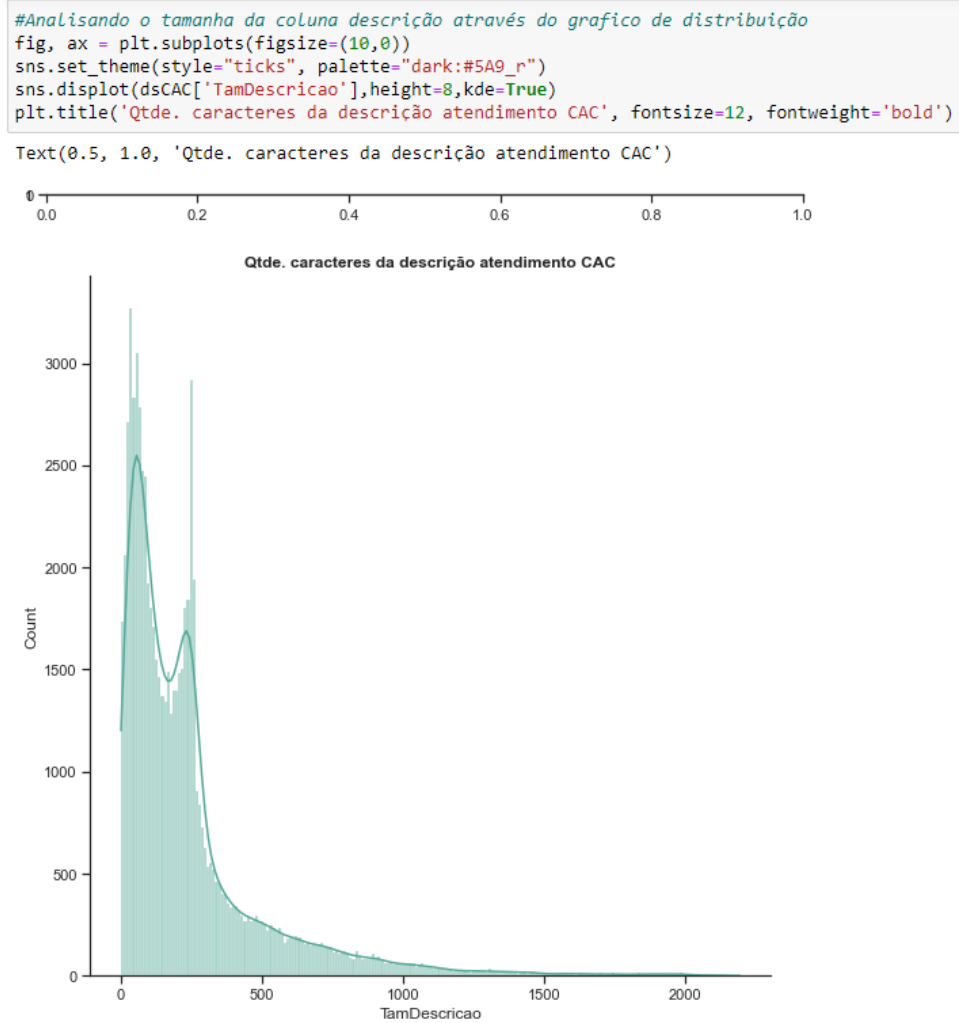


Figura 34 – Distribuição e concentração do "TamDescricao"

Utilizando o diagrama de caixa, fica mais fácil a interpretação dos dados, confirmando a concentração do tamanho da descrição até 400 caracteres. Os outliers estão acima de 600 caracteres (Figura 35 - Distribuição e concentração do "TamDescricao" através do diagrama de caixas e Figura 36 - Estatística do "TamDescricao"):

```
#Analisando o tamanho da coluna descrição através de diagrama de caixa
fig, aux = plt.subplots(figsize=(15,5))
sns.boxplot(x=dsCAC['TamDescricao'],notch=True, showcaps=True,palette="Set2",
            flierprops={"marker": "x"}, medianprops={"color": "red"})
plt.title('Distribuição tamanho da descrição', fontsize=14, fontweight='bold')
Text(0.5, 1.0, 'Distribuição tamanho da descrição')
```

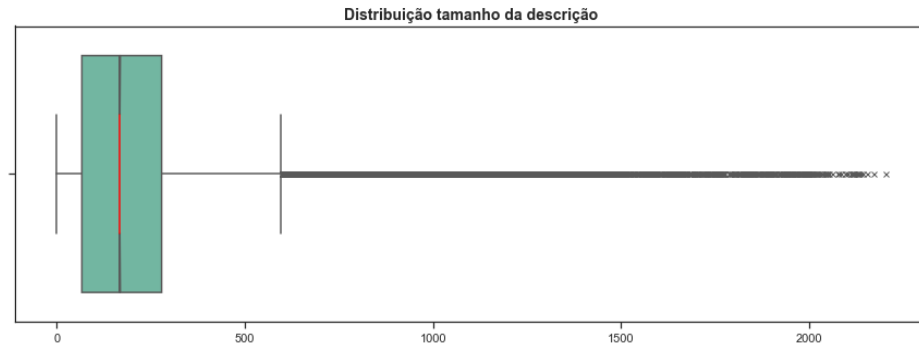


Figura 35 - Distribuição e concentração do "TamDescricao" através do diagrama de caixas

```
#Estatística geral da qtde caracteres da coluna Descricao
print("Estatística qtde caracteres:")
print("Mínima      : ", + dsCAC['TamDescricao'].min())
print("Máxima      : ", + dsCAC['TamDescricao'].max())
print("Média       : ", + int(dsCAC['TamDescricao'].mean()))
print("Mediana     : ", + int(np.median(dsCAC['TamDescricao'])))
print("Desvio Padrão : ", + int(np.std(dsCAC['TamDescricao'])))
```

```
Estatística qtde caracteres:
Mínima      : 0
Máxima      : 2197
Média       : 241
Mediana     : 166
Desvio Padrão : 271
```

Figura 36 - Estatística do "TamDescricao"

Baseado na dinâmica de uma CAC, podemos concluir que as descrições que estão como “outliers” (> 400 caracteres), foram atendimentos que necessitaram de mais iterações com o Cliente, ocasionando um registro maior de dados.

O dataset CAC possui 1.314 registros que não houve registro do contato do Cliente através das colunas “Descricao” e/ou “ExpectativaCliente” (Figura 37 – Amostra de registros do dataset CAC sem "Descricao" e/ou "ExpectativaCliente").

```
#fazendo uma cópia do dsCAC e retornando as descrições com tamanho = 0
dsCAC_Estatistica = dsCAC.copy()
dsCAC_Estatistica[dsCAC['TamDescricao']==0]
```

| | Classificacao | SubClassificacao | Data | Descricao | SegProdutoServico | GSSN | TamDescricao | Mes | Ano |
|-------|---------------|----------------------------------|------------|-----------|-------------------|-----------|--------------|-----|------|
| 623 | Solicitação | Questionamento do produto | 2023-07-04 | | Truck | NaN | 0 | 7 | 2023 |
| 2179 | Solicitação | Questionamento do Serviço | 2023-06-30 | | Truck | NaN | 0 | 6 | 2023 |
| 2250 | Solicitação | Questionamento do Serviço | 2023-06-26 | | Truck | NaN | 0 | 6 | 2023 |
| 2607 | Solicitação | Questionamento de vendas | 2023-06-08 | | Truck | NaN | 0 | 6 | 2023 |
| 2786 | Solicitação | Informação corporativa | 2023-04-27 | | Truck | NaN | 0 | 4 | 2023 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 60835 | Solicitação | Questionamento ao Concessionário | 2019-08-22 | | Truck | GS0018102 | 0 | 8 | 2019 |
| 62853 | Solicitação | Questionamento ao Concessionário | 2019-09-30 | | Truck | GS0021616 | 0 | 9 | 2019 |
| 62920 | Solicitação | Questionamento ao Concessionário | 2019-09-18 | | Truck | NaN | 0 | 9 | 2019 |
| 71016 | Solicitação | Questionamento ao Concessionário | 2019-05-16 | | Truck | NaN | 0 | 5 | 2019 |
| 71982 | Solicitação | Questionamento ao Concessionário | 2019-05-07 | | Truck | GS0093144 | 0 | 5 | 2019 |

1314 rows x 9 columns

Figura 37 – Amostra de registros do dataset CAC sem "Descricao" e/ou "ExpectativaCliente"

Consultando o campo “Descricao” com o maior tamanho de registro “2.203”. Somente um registro com o tamanho máximo (Figura 38 - Registro do dataset CAC com maior tamanho (qtde. de caracteres) da coluna "Descricao"):

```
dsCAC_Estatistica[dsCAC['TamDescricao']==2197]
```

| | Classificacao | SubClassificacao | Data | Descricao | SegProdutoServico | GSSN | TamDescricao | Mes | Ano |
|-------|---------------|---------------------|------------|---|-------------------|------------|--------------|-----|------|
| 12960 | Reclamação | Qualidade Pós Venda | 2022-05-04 | DESCRICAO CLIENTE ALISSON ENTROU EM CONTATO ... | Truck | GS00106242 | 2197 | 5 | 2022 |

```
print('Maior descrição : {}'.format(dsCAC_Estatistica.loc[12960]['Descricao']))
```

Maior descrição : DESCRICAO CLIENTE ALISSON ENTROU EM CONTATO PARA ABRIR UM PROTOCOLO PARA ACOMPANHAMENTO COM RELACAO A QUALIDADE DE REPARO CLIENTE ALEGA QUE LEVOU SEU VEICULO PARA CONCESSIONARIA MECASUL DE ENTRE IJUÍ - RS ENTRE O DIA // E // PARA REPARO SEGUNDO O CLIENTE O VEICULO COMEÇOU APRESENTAR FALHA NA TOMADA DE FORÇA E CHIADO DA SEXTA MARCHA ONDE ELE LEVOU O VEICULO PARA A CONCESSIONARIA EM QUESTAO VEICULO ENTAO FOI REPARADO E LIBERADO NO DIA // CLIENTE ENTAO COMEÇOU A RODAR COM SEU VEICULO ONDE COMEÇOU APRESENTAR A FALHA NOVAMENTE E ENTAO ELE SE DIRIGIU ATE A CONCESSIONARIA MEDIANERA RONDON NO RIO GRANDE DO SUL QUE SERIA UMA CONCESSIONARIA NAO AUTORIZADA PELA E COM ISSO SEGUNDO O CLIENTE O TECNICO DESSA CONCESSIONARIA INFORMOU QUE O VEICULO CONTINUA COM CHIADO NA SEXTA MARCHA E TAMBEM OCORREU UM ERRO POR PARTE DOS TECNICOS DA CONCESSIONARIA MECASUL ONDE OS MESMOS ACABARAM DANIFICANDO A TOMADA DE FORÇA DO VEICULO COMO O VEICULO NAO ESTA DENTRO DE UMA CONCESSIONARIA INFORMEI AO CLIENTE A SE DIRIGIR ATE UMA CONCESSIONARIA MAIS PROXIMA E POR TODO MAL ATENDIMENTO FORNECIDO POR PARTE DA CONCESSIONARIA MECASUL O CLIENTE INFORMA QUE VAI SE DIRIGIR PARA CONCESSIONARIA SAVAR DE PELOTAS - RS PARA QUE DE SEQUENCIA NOS DEVIDOS REPAROS CLIENTE ALEGA QUE NAO FOI FEITO NENHUM SERVICO NO VEICULO SEM SER POR PARTE DOS TECNICOS DAS CONCESSIONARIAS HOMOLOGADAS E QUE SO O DIGNOSTICO FOI FEITO PELO TECNICO DA CONCESSIONARIA RONDON COMO OS DIAGNOSTICOS FEITOS POR PARTE DAS CONCESSIONARIAS NAO HOMOLOGADAS PELAS NAO SAO VALIDOS SERIA IMPORTANTE QUE A CONCESSIONARIA SAVAR DE PELOTAS - RS FAÇA UM NOVO DIAGNOSTICO PARA IDENTIFICAR SE DE FATO A PANE SURTIU NOVAMENTE DEVIDO AOS SERVICOS PRESTADOS PELA CONCESSIONARIA MECASUL DE ENTRE IJUÍ - RS E COM ISSO REALIZAR OS REPAROS DENTRO DA GARANTIA REALIZEI A ABERTURA DO PROTOCOLO PARA ACOMPANHAMENTO PELO SEGUNDO NIVEL A PEDIDO DO CLIENTE POIS O MESMO ISISTIU QUE TIVESSE AUXILIO POR PARTE DA FAB EXPECTATIVA CLIENTE DESEJA CELERIDADE NAS TRATATIVAS DE SEU CASO POIS O VEICULO FICOU PARADO NA CONCESSIONARIA MECASUL ENTRE A DIAS E FOI TEORICAMENTE UM TEMPO JOGADO FORA POIS O VEICULO PERMANECE COM AS MESMAS FALHAS E DESEJA TAMBEM QUE OS FUTUROS

Figura 38 - Registro do dataset CAC com maior tamanho (qtde. de caracteres) da coluna "Descricao"

4.2 – Dataset RVAT

O dataset RVAT contém apenas 5 colunas de dados, as quais serão utilizadas para estatística geográfica quando relacionada com o dataset CAC através da coluna “GSSN” (Figura 39 - Informações do dataset RVAT).

```
dsRVAT.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 221 entries, 0 to 220
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    GSSN        221 non-null    object
1    Municipio  221 non-null    object
2    UF          221 non-null    object
3    Latitude    221 non-null    float64
4    Longitude   221 non-null    float64
dtypes: float64(2), object(3)
memory usage: 8.8+ KB

#formato do dataset
dsRVAT.shape

(221, 5)
```

Figura 39 - Informações do dataset RVAT

Distribuição da RVAT por ⁷UF (Figura 40 - Código para geração do gráfico de RVAT por UF e Figura 41 - Distribuição da RVAT por UF):

```
#Demonstração gráfica da qtde de RVAT por UF
fig, ax = plt.subplots(figsize=(15,5))
sns.countplot(dsRVAT['UF'],palette='pastel')
plt.title('Qtde. RVAT por UF',fontsize=16, fontweight='bold')
ax.set_xlabel('UF',fontsize=16, fontweight='bold')
ax.set_ylabel('Qtde. RVAT',fontsize=12, fontweight='bold')
plt.grid(False, axis='x')
plt.show()
```

Figura 40 - Código para geração do gráfico de RVAT por UF

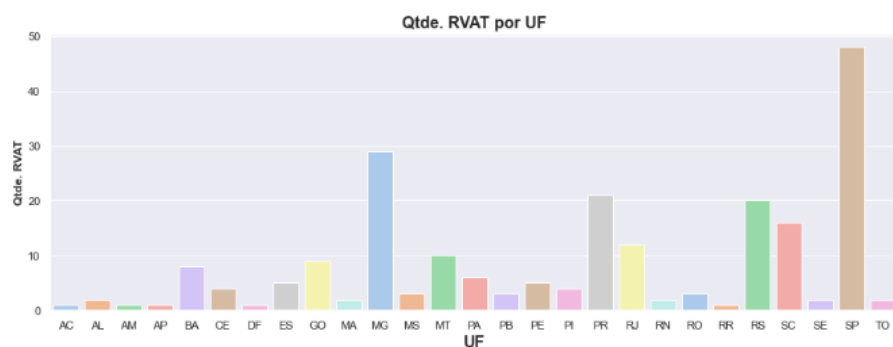


Figura 41 - Distribuição da RVAT por UF

A quantidade de RVAT por “UF” é reflexo da venda de veículos (emplacamentos) e serviços. As UF de SP, MG, PR, RS e SC (respectivamente) são os maiores polos comerciais do Brasil.

⁷ UF = Unidade da Federação (Estado)

A RVAT está localizada em 185 municípios, presente em todos as UF do Brasil (Figura 42 - Amostragem do dataset RVAT).

```
UF_Agrupado = dsRVAT.groupby(['UF', 'Municipio'])
UF_Agrupado.describe()
```

| UF | Municipio | Latitude | | | | | | | | Longitude | |
|-----|-----------------------|----------|------------|---------|------------|------------|------------|------------|------------|-----------|------------|
| | | count | mean | std | min | 25% | 50% | 75% | max | count | mean |
| AC | RIO BRANCO | 1.0 | -10.010980 | NaN | -10.010980 | -10.010980 | -10.010980 | -10.010980 | -10.010980 | 1.0 | -67.796630 |
| AL | ARAPIRACA | 1.0 | -9.749570 | NaN | -9.749570 | -9.749570 | -9.749570 | -9.749570 | -9.749570 | 1.0 | -36.635390 |
| | RIO LARGO | 1.0 | -9.499886 | NaN | -9.499886 | -9.499886 | -9.499886 | -9.499886 | -9.499886 | 1.0 | -35.811887 |
| AM | MANAUS | 1.0 | -3.047250 | NaN | -3.047250 | -3.047250 | -3.047250 | -3.047250 | -3.047250 | 1.0 | -60.025530 |
| AP | MACAPÁ | 1.0 | 0.040791 | NaN | 0.040791 | 0.040791 | 0.040791 | 0.040791 | 0.040791 | 1.0 | -51.074207 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| SP | SÃO JOÃO DA BOA VISTA | 1.0 | -21.998010 | NaN | -21.998010 | -21.998010 | -21.998010 | -21.998010 | -21.998010 | 1.0 | -46.815570 |
| | SÃO PAULO | 5.0 | -23.544283 | 0.04702 | -23.616435 | -23.567750 | -23.517441 | -23.510772 | -23.509019 | 5.0 | -46.672755 |
| | TABOÃO DA SERRA | 1.0 | -23.607206 | NaN | -23.607206 | -23.607206 | -23.607206 | -23.607206 | -23.607206 | 1.0 | -46.764405 |
| TO | ARAGUAÍNA | 1.0 | -7.170540 | NaN | -7.170540 | -7.170540 | -7.170540 | -7.170540 | -7.170540 | 1.0 | -48.213380 |
| | PALMAS | 1.0 | -10.249689 | NaN | -10.249689 | -10.249689 | -10.249689 | -10.249689 | -10.249689 | 1.0 | -48.314862 |

185 rows x 16 columns

Figura 42 - Amostragem do dataset RVAT

Através da base de dados de frota circulante, fornecido pelo ⁸Governo Federal e ⁹ANFAVEA, podemos verificar que a distribuição e quantidade da RVAT está diretamente proporcional a frota circulante. Essa proporcionalidade (frota circulante x RVAT) pode ser comprovada através da união dos datasets “FrotaCirculante” com o “RVAT” (Figura 43 - Frota circulante x RVAT):

```
#Leitura do dataset FrotaCirculante
dsFrota = pd.read_excel("FrotaCirculante.xlsx", sheet_name="FrotaCirculante")
dsFrota = dsFrota.sort_values(by=['FrotaCirculanteTotal'], ascending=False)

#unio os datasets "RVAT" e "FrotaCirculante"
UF_GSSN = dsRVAT.groupby(['UF']).count()
UF_GSSN = pd.DataFrame(UF_GSSN)
dsRVAT_FC = UF_GSSN.merge(dsFrota, how = 'inner', on = 'UF')
dsRVAT_FC.sort_values('FrotaCirculanteTotal', ascending=False, inplace=True)
dsRVAT_FC = pd.DataFrame(dsRVAT_FC)
dsRVAT_FC = dsRVAT_FC[['UF', 'GSSN', 'FrotaCirculanteTotal']]
dsRVAT_FC.columns = ['UF', 'Qtde_RVAT', 'FrotaCirculanteTotal']
def formatar(valor):
    return "{:,.2f}".format(valor)
dsRVAT_FC['FrotaCirculanteTotal'] = dsRVAT_FC['FrotaCirculanteTotal'].apply(formatar)
dsRVAT_FC
```

| | UF | Qtde_RVAT | FrotaCirculanteTotal |
|----|----|-----------|----------------------|
| 25 | SP | 48 | 404,459.00 |
| 10 | MG | 29 | 248,758.00 |
| 17 | PR | 21 | 152,340.00 |
| 22 | RS | 20 | 149,306.00 |
| 18 | RJ | 12 | 100,607.00 |
| 23 | SC | 16 | 81,921.00 |
| 4 | BA | 8 | 76,362.00 |
| 8 | GO | 9 | 65,869.00 |
| 7 | ES | 5 | 58,596.00 |

⁸ <https://www.gov.br/transportes/pt-br/assuntos/transito/conteudo-Senatran/frota-de-veiculos-2023>

⁹ Associação Nacional dos Fabricantes de Veículos Automotores: <https://anfavea.com.br>

Figura 43 - Frota circulante x RVAT

Nota: O estado do “RJ” possui frota circulante maior do que “SC” e RVAT menor. Isso é devido ao tamanho geográfico do “RJ” ser menor do que “SC”, demandando menos RVAT (Figura 44 - RVAT por UF):

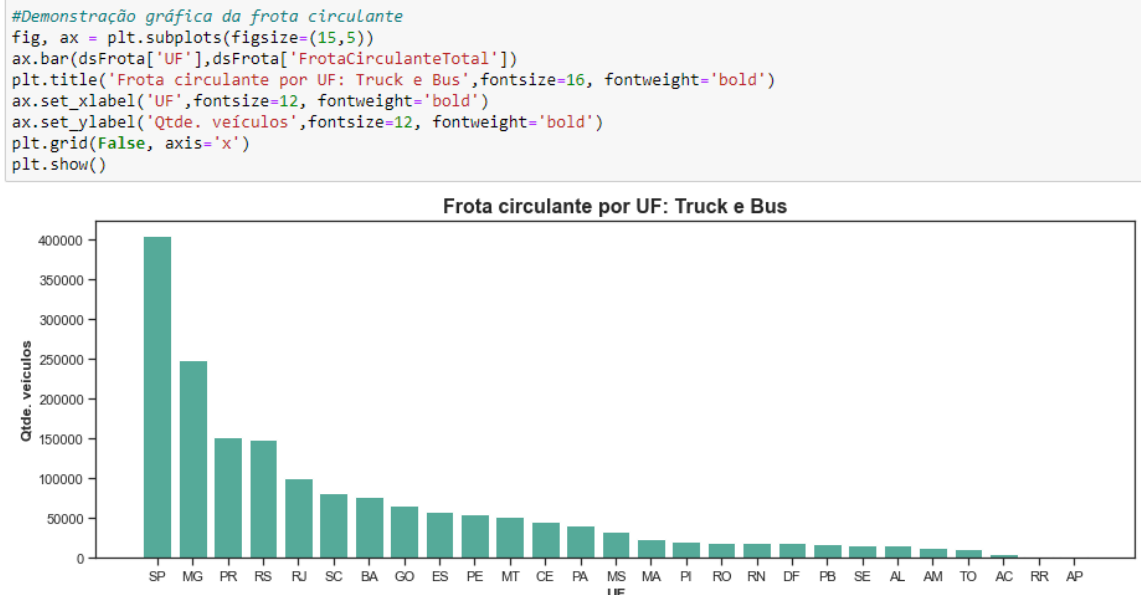


Figura 44 - RVAT por UF

4.3 – Relacionando os datasets CAC e RVAT

Os datasets CAC e RVAT possuem uma coluna em comum chamada de “GSSN”. Essa coluna é a conta comercial da RVAT e é única para cada RVAT (dataset RVAT).

Como não é mandatório a indicação da conta “GSSN” no ato do registro do atendimento via CAC (o Cliente poderá estar relatando ou solicitando apoio não ligado a um RVAT específico), nem todos os registros do dataset CAC possuem a conta “GSSN” informada.

A relação entre os datasets é (Figura 45 - Relacionamento entre os datasets RVAT e CAC):

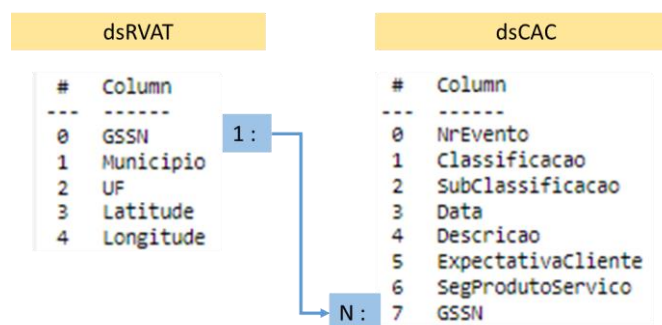


Figura 45 - Relacionamento entre os datasets RVAT e CAC

O dataset CAC possui 24.515 registros “não nulos” (onde consta o número da conta “GSSN”). Embora haja muitos registros sem a conta “GSSN” (45.895), utilizaremos essa coluna/relacionamento somente para estatística de quantos atendimentos CAC é feito por “UF”. O principal objetivo dessa dissertação é o “PLN”, onde trabalharemos com a coluna “Descricao” (Figura 46 - Verificando qtde de registros com “GSSN” nulo):

```
dsCAC.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 70410 entries, 0 to 73028
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Classificacao          70410 non-null  object
1   SubClassificacao       70410 non-null  object
2   Data                   70410 non-null  datetime64[ns]
3   Descricao              70410 non-null  object
4   SegProdutoServico      70410 non-null  object
5   GSSN                   24515 non-null  object
6   TamDescricao           70410 non-null  int64
7   Mes                    70410 non-null  int64
8   Ano                    70410 non-null  int64
dtypes: datetime64[ns](1), int64(3), object(5)
memory usage: 7.4+ MB
```

```
dsCAC.isnull().sum()

Classificacao          0
SubClassificacao       0
Data                   0
Descricao              0
SegProdutoServico      0
GSSN                   45895
TamDescricao           0
Mes                    0
Ano                    0
dtype: int64
```

Figura 46 - Verificando qtde de registros com “GSSN” nulo

Fazendo o relacionamento entre os datasets RVAT e CAC, utilizando o tipo de união “inner”, retornará somente os registros onde o valor da coluna “GSSN” estiver presente em ambos datasets (Figura 47 - Dados gerais após relacionamento dos datasets RVAT e CAC):

```
#relacionando os datasets dsRVAT e dsCAC
dsRVAT_CAC = dsRVAT.merge(dsCAC, how = 'inner', on = 'GSSN')

dsRVAT_CAC.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 24297 entries, 0 to 24296
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   GSSN                   24297 non-null object
1   Municipio              24297 non-null object
2   UF                     24297 non-null object
3   Latitude                24297 non-null object
4   Longitude               24297 non-null object
5   Classificacao           24297 non-null object
6   SubClassificacao        24297 non-null object
7   Data                   24297 non-null datetime64[ns]
8   Descricao               24297 non-null object
9   SegProdutoServico       24297 non-null object
10  TamDescricao            24297 non-null int64
11  Mes                     24297 non-null int64
12  Ano                     24297 non-null int64
dtypes: datetime64[ns](1), int64(3), object(9)
memory usage: 3.1+ MB
```

Figura 47 - Dados gerais após relacionamento dos datasets RVAT e CAC

Amostragem dos dados após a união dos datasets (Figura 48 - Amostragem dos dados após o relacionamento dos datasets RVAT e CAC):

```
display(dsRVAT_CAC)
```

| | GSSN | Municipio | UF | Latitude | Longitude | Classificacao | SubClassificacao | Data | Descricao | SegProdutoServico | TamDescri |
|-------|-----------|------------|-----|-----------|-----------|---------------|----------------------------------|------------|---|-------------------|-----------|
| 0 | GS0003147 | RIO BRANCO | AC | -10,01098 | -67,79663 | Solicitação | Questionamento ao Concessionário | 2023-06-30 | delis.bezerra@acrediesel.com.br Boa tarde ... | Truck | |
| 1 | GS0003147 | RIO BRANCO | AC | -10,01098 | -67,79663 | Solicitação | Questionamento ao Concessionário | 2023-06-24 | GS Boa tarde Solicito por gentileza que o... | Truck | |
| 2 | GS0003147 | RIO BRANCO | AC | -10,01098 | -67,79663 | Solicitação | Questionamento ao Concessionário | 2023-06-27 | - ATUALIZACAO CADASTRAL | Truck | |
| 3 | GS0003147 | RIO BRANCO | AC | -10,01098 | -67,79663 | Reclamação | Qualidade Pós Venda | 2023-06-28 | SR FLAUBERTH EMAIL – FLAUBERTH.BASTOS@FOG... | Truck | 1 |
| 4 | GS0003147 | RIO BRANCO | AC | -10,01098 | -67,79663 | Solicitação | Questionamento ao Concessionário | 2023-07-12 | ACESSO MT E TCAS@DEALER | Truck | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 24292 | GS0018106 | ARAGUAÍNA | TO | -7,17054 | -48,21338 | Agendamento | Agendamento reparo | 2019-04-29 | Sr. Anderson Fone: - Agendado: // às : hs Conc... | Truck | |
| 24293 | GS0018106 | ARAGUAÍNA | TO | -7,17054 | -48,21338 | Agendamento | Agendamento reparo | 2019-04-29 | Sr. Anderson Fone: - Agendado: // às : hs Conc... | Truck | |
| 24294 | GS0018106 | ARAGUAÍNA | TO | -7,17054 | -48,21338 | Agendamento | Agendamento reparo | 2019-04-29 | Sr. Anderson Fone: - Agendado: // às : hs Conc... | Truck | |
| 24295 | GS0018106 | ARAGUAÍNA | TO | -7,17054 | -48,21338 | Agendamento | Agendamento reparo | 2019-04-29 | Sr. Anderson Fone: - Agendado: // às : hs Conc... | Truck | |
| 24296 | GS0018106 | ARAGUAÍNA | TO | -7,17054 | -48,21338 | Solicitação | Questionamento ao Concessionário | 2019-04-16 | CONC. QUESTIONA SOBRE PLANO DE MANUTENÇÃO aut... | Truck | |

24297 rows x 13 columns

Figura 48 - Amostragem dos dados após o relacionamento dos datasets RVAT e CAC

Com os datasets unidos, podemos gerar um gráfico demonstrando a quantidade de atendimentos CAC por UF. Utilizamos uma função para sumarizar os atendimentos em um dicionário de dados, posteriormente transformarmos esse dicionário em duas listas (listaUF e

listaQtde) e fizemos a plotagem em um gráfico de barras verticais (Figura 49 - Código para plotagem do gráfico atendimento CAC por UF e Figura 50 - Gráfico de atendimento CAC por UF):

```
#Gerando gráfico de barras horizontais para demonstrar
#qtde de atendimentos via CAC por UF
dsUF_Atend = dsRVAT_CAC.groupby(by=['UF'])['GSSN'].count().reset_index()
dsUF_Atend = dsUF_Atend.sort_values(by=['GSSN'])
plt.figure(figsize=(10, 10))
plt.title('Atendimentos CAC por UF',fontsize=16, fontweight='bold')
plt.xlabel('Qtde',fontsize=16,fontweight='bold')
plt.ylabel('UF',fontsize=16, fontweight='bold')
plt.barh(dsUF_Atend['UF'], dsUF_Atend['GSSN'], align='center')
plt.grid(True, axis='x')
```

Figura 49 - Código para plotagem do gráfico atendimento CAC por UF

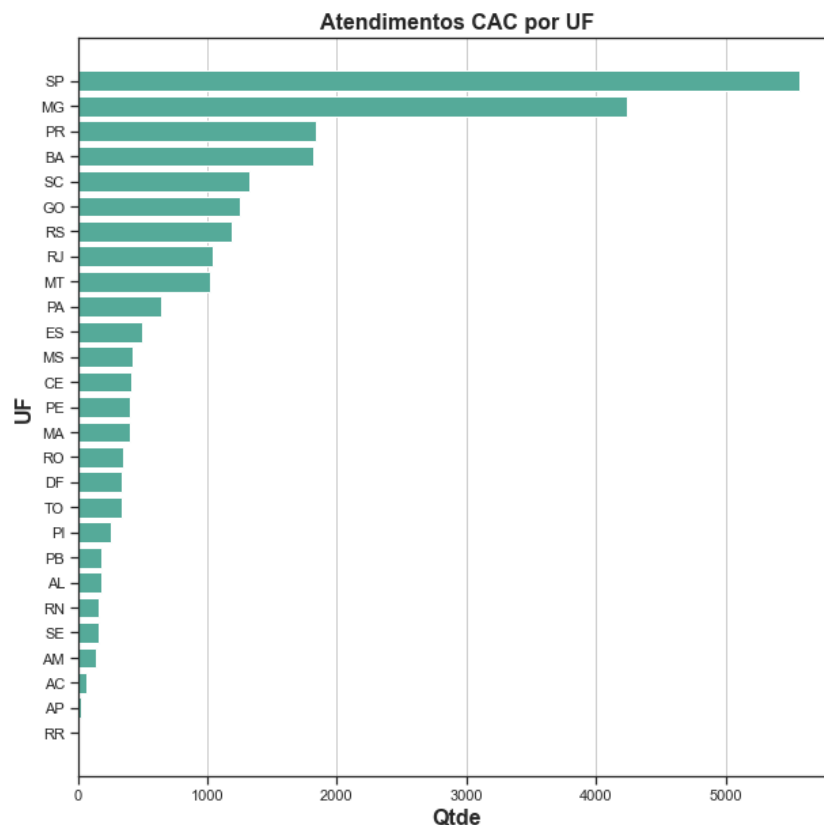


Figura 50 - Gráfico de atendimento CAC por UF

Uma outra forma de analisarmos a concentração de atendimentos CAC por município é através do mapa de calor, utilizando como base a latitude e longitude do município onde a RVAT está localizada (24.297 atendimentos CAC que possuem conta GSSN e consequentemente a latitude e longitude pelo relacionamento entre os datasets RVAT e CAC). Para essa análise, utilizaremos a biblioteca “Folium” para Python (Figura 51 - Código para plotagem do mapa de calor através da biblioteca "Folium"):

```
##### Gerando gráfico de calor através do Folium
#importando a biblioteca Folium
import folium
from folium import plugins
from folium.plugins import HeatMap
from branca.colormap import LinearColormap # Create a colormap instance
import json

dfLatLon = pd.DataFrame(dsRVAT_CAC, columns=['Latitude','Longitude']).values.tolist()
dsUF_Atend.rename(columns={'GSSN': 'QtdeAtend'}, inplace = True)

#definindo ponto inicial do mapa, zoom inicial e demais parametros
mapa = folium.Map(width='100%',height='100%', location=[-15.77972, -47.92972],
                  zoom_start=4.45, tile='Stamen Terrain')
colormap = LinearColormap(colors=['white', 'green','blue','yellow', 'red'],
                          vmin=dsUF_Atend['QtdeAtend'].min(), vmax=dsUF_Atend['QtdeAtend'].max())
colormap.caption = 'Índice de atendimento CAC nacional'
colormap.add_to(mapa)

#gerando o mapa de calor baseado na qtde de atendimentos CAC por Latitude e Longitude
HeatMap(dfLatLon, radius = 15).add_to(mapa)

# Criando o círculo e os tooltips com informações
dsMun_Atend= dsRVAT_CAC.groupby(by=['UF','Município','Latitude','Longitude'])['GSSN'].count().reset_index()
dsMun_Atend.rename(columns={'GSSN': 'QtdeAtend'}, inplace = True)

for i in range(0, len(dsMun_Atend)):
    folium.Circle(
        location = [dsMun_Atend.iloc[i]['Latitude'], dsMun_Atend.iloc[i]['Longitude']],
        color = '#000000',
        fill = '#00A1B3',
        tooltip = '<li><b> Município: ' + str(dsMun_Atend.iloc[i]['Município']) +
        '<li><b> Estado: ' + str(dsMun_Atend.iloc[i]['UF']) +
        '<li><b> Qtde. atendimentos: ' + str(int(dsMun_Atend.iloc[i]['QtdeAtend'])),
        radius = 10
    ).add_to(mapa)
mapa
```

Figura 51 - Código para plotagem do mapa de calor através da biblioteca "Folium"

Através do gráfico de calor identificamos que a concentração dos atendimentos está nas regiões Sudeste Sul e Nordeste consecutivamente. Ao passar o mouse sobre o ponto preto, recebemos as informações daquela cidade (Figura 52 - Mapa de calor atendimento CAC nacional):

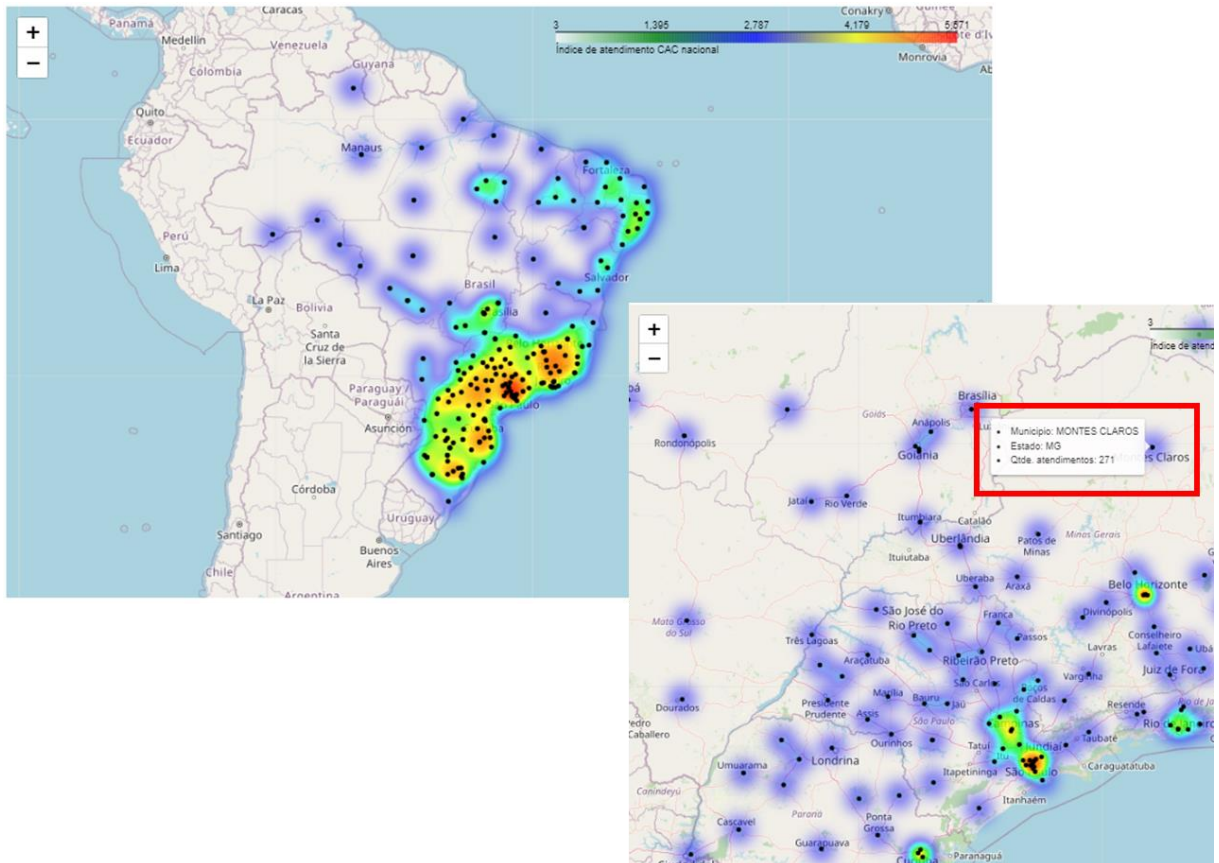


Figura 52 - Mapa de calor atendimento CAC nacional

5. Criação de Modelos de Machine Learning

Como mencionado no item **1.3. Objetivos**, utilizaremos o modelo de NLP nessa dissertação.

Para usarmos um modelo estatístico ou de deep learning em NLP, precisamos de features: informações mensuráveis acerca de algum fenômeno, ou seja, uma forma estruturada de armazenar informações. Porém, textos são um tipo de dado não estruturado (não organizado de uma maneira pré-definida, fixa), assim, é difícil para o computador entendê-los e analisá-los. Por isso, realizamos a chamada *feature extraction*, ou seja, **transformamos o texto em uma informação numérica** de modo que seja possível utilizá-lo para alimentar um modelo. Uma das maneiras mais populares e simples de fazer isso é com Bag of Words (BoW) e TF-IDF.

5.1 – Bag Of Words

BoW é uma forma de representar o texto de acordo com a ocorrência das palavras nele. Traduzindo para o português, o “saco de palavras” recebe esse nome porque não leva em conta a ordem ou a estrutura das palavras no texto, apenas se ela aparece ou a frequência com que aparece nele.

Por exemplo, se a palavra “Manutenção” aparece muito num texto, ela se torna mais central e importante para a máquina. Portanto, BoW pode ser um ótimo método para determinar as palavras significativas de um texto com base no número de vezes que ela é usada.

Para gerar um modelo de bag of words aplicaremos alguns processos que serão descritos no capítulo: 5.3 – Implementação NLP.

5.2 - Term Frequency - Inverse Document Frequency (TF-IDF)

Trata-se de medidas estatísticas para medir o quão importante uma palavra é em um documento (texto), assim como BoW, mas com algumas diferenças.

Com ele, podemos perceber a importância de uma palavra por meio de uma pontuação, o TF-IDF de uma palavra em um texto é feito multiplicando duas métricas diferentes:

- Term Frequency (TF - a frequência do termo), que mede a frequência com que um termo ocorre num documento;

- Inverse Document Frequency (IDF - inverso da frequência nos documentos), que mede o quão importante um termo é no contexto de todos os documentos (Figura 53 - Cálculo do TF-IDF):

$$TFIDF = TF \times IDF$$

ou

$$TFIDF = \frac{\text{Nº DE VEZES QUE UMA PALAVRA APARECE EM UM DOCUMENTO}}{\text{Nº DE PALAVRAS DO DOCUMENTO}} \times \log \left(\frac{\text{TOTAL DE DOCUMENTOS}}{\text{Nº DE DOCUMENTOS COM O RESPECTIVO TERMO}} \right)$$

Figura 53 - Cálculo do TF-IDF

Para o TF-IDF, quanto mais frequente uma palavra é em seu documento, mais importante ela é no contexto. Entretanto, isso depende da repetição dela ao longo de todos os documentos que estão sendo analisados.

Para a implementação de TF-IDF precisamos seguir alguns processos semelhantes ao BoW. Utilizaremos técnicas baseadas na biblioteca scikit-learn e nltk. A Figura 54 - Fluxo tratamento texto para implementação de NLP representa o processo utilizado para o tratamento do texto:

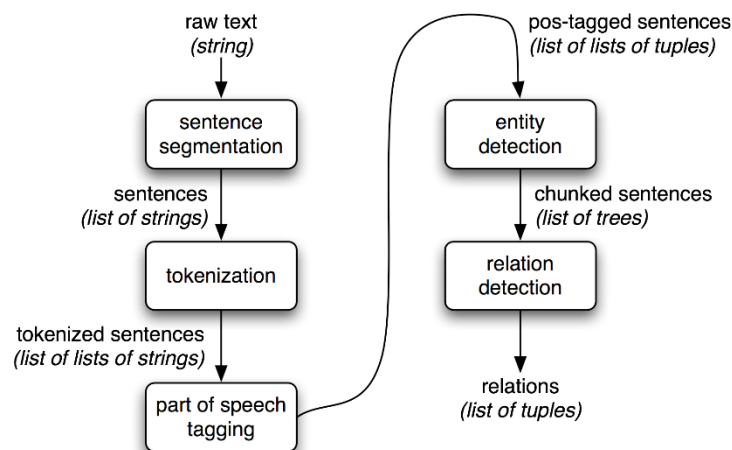


Figura 54 - Fluxo tratamento texto para implementação de NLP

5.3 – Implementação NLP

5.3.1 – Seleção dos dados

Utilizaremos o dataset CAC, mais precisamente os dados constantes da coluna “Descricao”. Nessa coluna, constam em forma descritiva, tudo o que foi reportado pelo cliente

e atendente do CAC no momento do contato e em todos os outros contatos necessários até o encerramento do chamado (protocolo).

5.3.2 – Pré-processamento do texto

O pré-processamento é uma etapa da preparação do texto com o objetivo de reduzir o tamanho do vocabulário (conjunto de palavras que ocorrem em todos os textos do dataset) e simplificar algumas formas lexicais para que o nosso algoritmo consiga extrair informações relevantes e tenha um desempenho melhor.

5.3.1.1 Etapas do pré-processamento do texto

a) Expressões regulares

Expressões regulares é uma sequência de caracteres utilizada para encontrar ou substituir padrões em uma string. Usaremos deste recurso para manter apenas os caracteres que representam letras do alfabeto.

Para trabalhar com Expressões Regulares em Python, vamos importar a biblioteca “re” e utilizar o método `.findall()`. Nele, passamos o padrão a ser procurado e a string em que ele deve buscar esta informação.

Para o Python as palavras como “oi” e “Oi” são interpretadas como palavras diferentes pelo sistema. Converteremos os texto para minúscula utilizando o método `.lower()`.

b) Tokenização

Outra etapa importante no pré-processamentos de PLN é a tokenização. Tokenização significa pegar cada uma das palavras do texto (tokens) e armazená-las em uma lista. Para fazer isso, poderíamos ter usado tokenizadores de bibliotecas como NLTK, entretanto, o método `.findall()` já nos retorna uma lista com as palavras do texto. Por isso, iremos pular esta etapa.

c) Stopwords

Stopwords são palavras que, apesar de muito frequentes, carregam pouca relevância semântica. Entre elas, podemos encontrar artigos como “o” e “uma”, ou preposições como “de” e “em”, entre outras palavras frequentes no idioma. Para removê-las do texto, utilizamos uma lista de stopwords em português disponível na biblioteca NLTK.

Em português, não encontramos nessa lista algumas formas contraídas de verbos ou preposições, como “tá” e “pra”. Sendo assim, adicionamos manualmente essas palavras.

d) Lematização

Nesta etapa, iremos passar o texto por uma simplificação lexical. Para isso, há dois processos possíveis: Stemização e Lematização. Stemização refere-se ao processo de reduzir as palavras flexionadas à sua raiz. Lematização é responsável por representar as palavras através do infinitivo para os verbos, e do masculino singular no caso de substantivos e adjetivos.

No caso da Stemização, por retornar apenas a raiz da palavra, é possível que o resultado seja uma forma não dicionarizada desta. Como iremos utilizar as palavras para a nossa análise, utilizaremos a Lematização.

Utilizaremos o lematizador da biblioteca spaCy. Entretanto, ainda existem formas verbais que não são detectadas por essa ferramenta em português. Sendo assim, adicionamos manualmente também uma lista com alguns desses verbos.

Antes de iniciarmos o tratamento do texto (“Descricao”), vamos retirar os registros cujos tamanhos da descrição são ¹⁰outliers (veja: Figura 35 - Distribuição e concentração do “Tam-Descricao” através do diagrama de caixas). Utilizaremos o método Tukey para definição dos outliers (método utilizado pelo gráfico bloxplot): definição dos limites inferior e superior a partir do interquartil (IQR) e dos primeiros (Q1) e terceiros (Q3) quartis (Figura 55 - Método Turkey para identificação de outliers):

¹⁰ Outliers são dados discrepantes em relação a um conjunto de dados. Este tipo de dado pode trazer distorções em nossas análises.

```

# removendo os outliers do dataset dsCAC
#dataset completo "dsCAC"
print("Dataset completo: ", dsCAC.shape)

#limite inferior do quartil
Q1 = np.percentile(dsCAC['TamDescricao'], 25, interpolation = 'midpoint')

#limite superior do quartil
Q3 = np.percentile(dsCAC['TamDescricao'], 75, interpolation = 'midpoint')

#interquartil
IQR = Q3 - Q1

#removendo os registros outliers (inferiores e superiores)
dsCAC_SemOutliers=dsCAC[(dsCAC.TamDescricao>=int(Q1-1.5*IQR)) & (dsCAC.TamDescricao<=int(Q3+1.5*IQR))]
dsCAC_SemOutliers = pd.DataFrame(dsCAC_SemOutliers)

#dataset excluindo os outliers
print("Novo dataset: ", dsCAC_SemOutliers.shape)

Dataset completo: (70410, 9)
Novo dataset: (63770, 9)

```

Figura 55 - Método Turkey para identificação de outliers

Podemos verificar que houve redução de 6.589 registros do dataset original que são os outliers.

Aplicaremos a função “limpa_texto” para processar todas as etapas que descrevemos acima:

- Deixar tudo em minúsculo (para que o computador não considere ‘brasil’ e ‘Brasil’ como palavras diferentes)
- Filtrar apenas letras (removendo pontuações, símbolos, etc.)
- Remover stopwords (palavras que se repetem muito no texto, mas adicionam pouca informação, como ‘de’)
- Lematizar (simplificação lexical, por exemplo, passando os verbos para o infinitivo e passando substantivos e adjetivos para o masculino singular)

Nota: devido ao erro “WordCloud Only Supported for TrueType fonts” foi necessário a instalação da biblioteca “freetype” (high-level Python API) para corrigir esse problema (Figura 56 - Instalação da biblioteca “freetype”):

```

#instalando a biblioteca freetype para evitar erro:
#WordCloud Only Supported for TrueType fonts
!pip install freetype-py

```

Figura 56 - Instalação da biblioteca “freetype”

Aplicando a função “limpa_texto” à coluna “Descricao” (Figura 57 - Tratamento do texto com a função “limpa_texto”):


```
def limpa_texto(descricao):

    #removendo todos os caracteres que não são ASCII e substituindo pelo caracter ASCII mais próximo
    descricao_ = unidecode.unidecode(descricao)

    # Remover caracteres que não são letras e tokenização
    descricao_ = re.findall(r'\b[A-zÃ-úü]+\b', descricao_.lower())

    #Remover stopwords
    stopwords = nltk.corpus.stopwords.words('portuguese')

    #Adicionando stopwords que não estão na lista original
    stopwords.append("")
    stopwords.append("area")
    stopwords.append("aberta")
    stopwords.append("abraco")
    stopwords.append("abraço")
    stopwords.append("veiculo")
    stopwords.append("contato")
    stopwords.append("atraves")
    stopwords.append("atrave")
    stopwords.append("cliente")
    stopwords.append("br")
    stopwords.append("km")
    stopwords.append("dia")
    stopwords.append("informar")
    stopwords.append("concessionario")
    stopwords.append("concessionaria")
    stopwords.append("conc")
    stop = set(stopwords)

    meaningful_words = [w for w in descricao_ if w not in stop]
    meaningful_words_string = " ".join(meaningful_words)

    #Instanciando o objeto spacy
    spc_descricao_ = spc_pt(meaningful_words_string)

    #Lemmização
    tokens = [token.lemma_ if token.pos_ == 'VERB' else str(token) for token in spc_descricao_]
    tokens_ = tokens

    #tratamento específico para o verbo "ir"
    ir = ['vou', 'vais', 'vai', 'vamos', 'ides', 'vão']
    tokens = ['ir' if token in ir else str(token) for token in tokens]

    return " ".join(tokens)
```

Figura 57 - Tratamento do texto com a função "limpa_texto"

Vamos gerar um arquivo externo com o resultado do tratamento do texto "Descricao". Esse arquivo será do tipo ".csv" e terá o nome "dsCAC_preprocessado.csv" (Figura 58 - Gerando arquivo externo com texto "Descricao" processado):

```
#Gerando arquivo do dataset pre-processado
dsCAC_SemOutliers.to_csv('dsCAC_preprocessado.csv', index= False, columns= ['Classificacao', 'SubClassificacao', 'Data', 'Descricao'])
```

Figura 58 - Gerando arquivo externo com texto "Descricao" processado

Filtrando o dataset CAC para retornar somente os registros que possuem dados na coluna "Descricao" (Figura 59 - Filtrando o dataset CAC):

```
#filtrando somente os registros que possuem dados na coluna "Descricao"
dsCAC_SemOutliers = dsCAC_SemOutliers[dsCAC_SemOutliers['Descricao'].notnull()]
```

Figura 59 - Filtrando o dataset CAC

5.3.1.2 Geração do BoW e TF-IDF

Chegamos no momento da criação do BoW e a utilização do TF-IDF. Utilizaremos a biblioteca scikit e o método “CountVector” que converte o texto em uma matriz de “token” contabilizado. Como já fizemos o tratamento do texto da coluna “Descricao” através da função “limpa_texto” e aplicamos o resultado na própria coluna, não será necessário chamar a função “limpa_texto” novamente através do método “analyser” (Figura 60 - Aplicação dos modelos BoW e TF-IDF):

- min_df = 5 : desconsiderar palavra que apareça em menos de 5 registros
- max_df = .50 : desconsiderar palavra que apareça em mais de 50% dos registros
- max_features=None : será analisado todas as “features”

```
# Importando o CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer
#converte o texto em uma matrix de tokens contabilizados/contados
#min_df=5 --> desconsiderar palavra que apareça em menos de 5 registros
#max_df = .50 --> desconsiderar palavra que apareça em mais de 50% dos registros
#max_features=None --> será analisado todas as “fetures”
vectorizer = CountVectorizer(min_df=5, max_df=.50, max_features=None)
bow_vector = vectorizer.fit_transform(dsCAC_SemOutliers['Descricao'])
feature_names = vectorizer.get_feature_names()

#Transformando uma matrix contabilizada/contada em uma representação tf/tf-idf normalizada
from sklearn.feature_extraction.text import TfidfTransformer
tfidf_transformer = TfidfTransformer().fit(bow_vector)

#transformando BoW em corpus TF-IDF
tfidf_vector = tfidf_transformer.transform(bow_vector)
```

Figura 60 - Aplicação dos modelos BoW e TF-IDF

Após a aplicação do modelo BoW e TF-IDF, obtivemos a matriz esparsa “tfidf_vector” com 62.412 registros processados (linhas), gerando 8.443 colunas (terms/features: não houve limite de radicais de palavras processadas) e 830.766 ocorrências (Figura 61 - Resultado do processamento do modelo BoW e TF-IDF **Erro! Fonte de referência não encontrada.**):

```
tfidf_vector
```

```
<62412x8443 sparse matrix of type '<class 'numpy.float64'>'
  with 830766 stored elements in Compressed Sparse Row format>
```

Figura 61 - Resultado do processamento do modelo BoW e TF-IDF

Utilizaremos o recurso chamado de “nuvem de palavras” ou “Word Cloud” para uma análise visual do resultado dos termos mais frequentes. A Figura 62 - Código Python para geração da nuvem de palavras contém o código Python para a geração da “nuvem de palavras” (Figura 62 - Código Python para geração da nuvem de palavras):

```
#Nuvem de palavras com "Descricao" do dataset dsCAC_SemOutliers
from wordcloud import WordCloud, ImageColorGenerator
from PIL import Image

# definindo uma imagem como máscara
mask1 = np.array(Image.open("MapaBrasil.png"))

font_path = "GoldUnderTheMud-Regular.ttf" #utilizando fonte não padrão
text = ' '.join(texto for texto in dsCAC_SemOutliers['Descricao'])

wordcloud = WordCloud(background_color="white",width=1000, height=500,
                      font_path=font_path, colormap="copper",
                      collocations = False,mask=mask1, min_word_length=3,max_words=200)
wordcloud.generate(text)
plt.figure(figsize=(20,15))
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
```

Figura 62 - Código Python para geração da nuvem de palavras

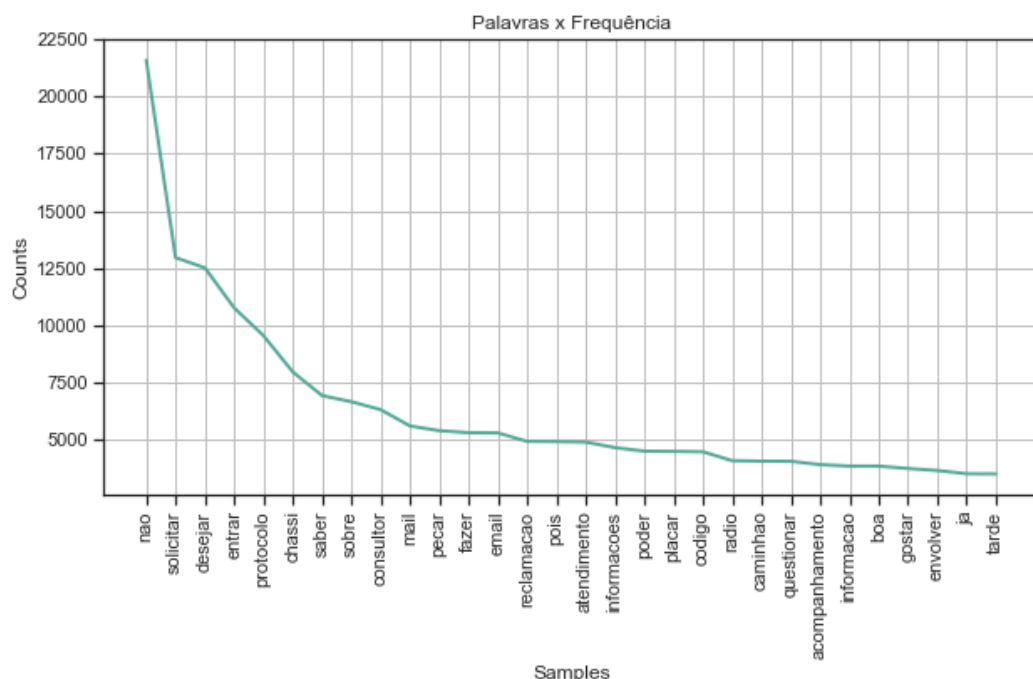
Abaixo segue o quadro com a nuvem de palavras geradas (Figura 63 - Nuvem de palavras utilizando o formato do mapa do Brasil):


```
from nltk.tokenize import word_tokenize
text_ = word_tokenize(text)
from nltk.probability import FreqDist
fdist = FreqDist(text_)
print(fdist.most_common(30))
```

```
[('nao', 21599), ('solicitar', 12955), ('desejar', 12510), ('entrar', 10759), ('protocolo', 9543), ('chassi', 7956), ('saber', 6921), ('sobre', 6652), ('consultor', 6306), ('mail', 5592), ('pecar', 5385), ('fazer', 5298), ('email', 5287), ('reclamacao', 4918), ('pois', 4908), ('atendimento', 4886), ('informacoes', 4644), ('poder', 4490), ('placar', 4478), ('codigo', 4462), ('radio', 4071), ('caminhao', 4052), ('questionar', 4045), ('acompanhamento', 3900), ('informacao', 3836), ('boa', 3835), ('gostar', 3732), ('envolver', 3643), ('ja', 3503), ('tarde', 3495)]
```

Figura 64 - 30 palavras mais frequentes

```
plt.figure(figsize=(10,5))
fd = nltk.FreqDist(text_)
fd.plot(30,title = "Palavras x Frequência",cumulative=False)
```



```
<AxesSubplot:title={'center':'Palavras x Frequência'}, xlabel='Samples', ylabel='Counts'>
```

Figura 65 - Gráfico distribuição das 30 palavras mais frequentes

Fazendo uma análise geral, a nuvem gerada condiz com o grupo de palavras mais presentes em um atendimento CAC: “não” (geralmente associado a reclamação), “informar” (solicitando informações do atendimento), “solicitar” (relacionado a algum pedido), “desejar” (relacionado a necessidade de saber a situação do chamado), “protocolo” (relacionado ao chamado CAC), “chassi” (identificação do veículo), “reclamação”, “rádio e código” (referente a perda do código para ativar o rádio) e etc.

5.3.1.3 Análise e aplicação da redução da dimensionalidade

Outro processo importante na modelagem dos dados para aplicação de “Machine Learning” é a redução da dimensionalidade dos dados (features), conhecido como a “maldição

da dimensionalidade”. A alta dimensionalidade dos dados é um ponto bastante importante quando estamos desenvolvendo um projeto de “Machine Learning”, pois modelos matemáticos sofrem bastante influência dos aumentos ou reduções das dimensões dos dados, impactando diretamente no seu desempenho.

A redução da dimensionalidade (features) é o processo de reduzir o número de variáveis aleatórias que serão inseridas em um modelo para treino. Imaginamos um dataset com centenas de colunas (no nosso estudo, são 8.947 colunas/features), a redução de dimensionalidade traria as dimensões para um número mais fácil de se trabalhar, algumas poucas dezenas por exemplo. Seria como converter uma esfera de três dimensões para um círculo de duas dimensões (Figura 66 - Representação da redução da dimensionalidade):

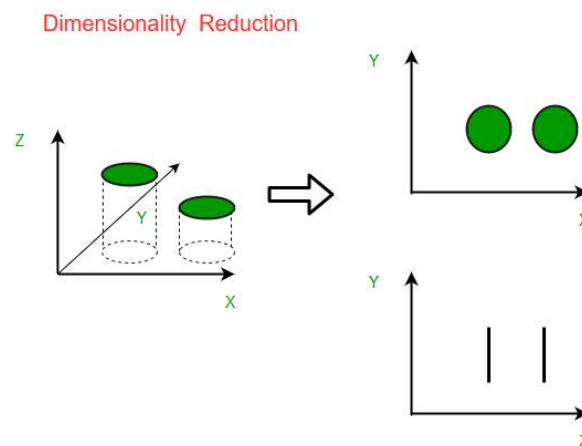


Figura 66 - Representação da redução da dimensionalidade

A redução de dimensionalidade é capaz de simplificar modelos, reduzir o tempo de treino e reduzir o overfitting.

O overfitting ocorre quando um modelo treina os dados "bem demais". Ou seja, o modelo entende perfeitamente os dados utilizados no treino. Funcionando bem até para os ruídos ou dados comprometidos do sistema, isso faz com que tenha um resultado excelente para os dados de treino, mas tenha um resultado ruim com dados novos.

Redução de Dimensionalidade é muito útil para aprendizado não supervisionado. Nesse tipo de aprendizado, inferências são extraídas das features sem saber quais os seus rótulos ou classes. É utilizado para explorar padrões ocultos ou agrupamentos de dados nos dados.

No Scikit-Learn temos algumas técnicas para redução de dimensionalidade:

- Principal component analysis (PCA)
- Latent Semantic Indexing (LSA/LSI), também conhecido como Truncated SVD (Singular Value Decomposition)
- Linear Discriminant Analysis (LDA)
- Isomap

Nota: PCA e LDA são métodos lineares e o Isomap é um método não-linear.

Utilizaremos a técnica de redução de dimensionalidade ¹¹Truncated SVD / LSA.

O que é o Truncated SVD / LSA?

O SVD é uma técnica de redução da dimensionalidade que reduz a matriz em componentes para simplificar o cálculo.

Foi feita uma tentativa de redução da dimensionalidade utilizando 5.000 componentes, mas devido à falta de memória para processamento, reduzimos para 3.000 componentes. Como não houve uma estabilidade na variância, não aplicaremos a redução da dimensionalidade (Figura 67 - Tentativa de redução da dimensionalidade utilizando 3.000 componentes).

¹¹ <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>

```
#redução da dimensionalidade em 3.000 componentes
from sklearn.decomposition import TruncatedSVD
svd=TruncatedSVD(n_components=3000)
svd_vec=svd.fit_transform(tfidf_vector)

explained_variances=[i/np.sum(svd.explained_variance_ratio_) for i in svd.explained_variance_ratio_]
variances=[]
temp=0
for i in explained_variances:
    temp=temp+i
    variances.append(temp)
plt.plot(variances,label='Explained Variances')
plt.xlabel("explained Variances")
plt.show()
```

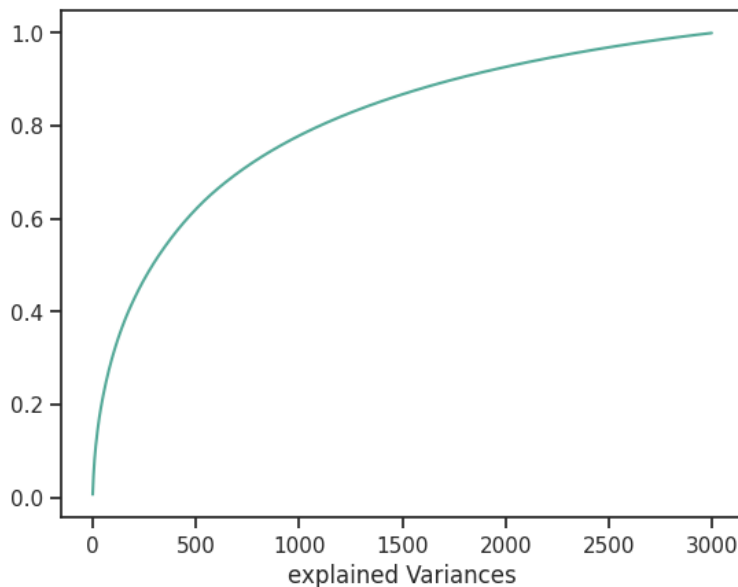


Figura 67 - Tentativa de redução da dimensionalidade utilizando 3.000 componentes

5.3.1.4 K-Means

Por padrão, o K-Means utiliza a quantidade de clusters igual a 10 (dez). Existem alguns recursos para tentarmos encontrar a quantidade ideal de cluster baseado nos dados que estamos processando. Dentro os recursos disponíveis, temos o “método da curva de cotovelo” e o método de “Silhouette”.

5.3.1.4.1 Método curva de cotovelo

O “método da curva de cotovelo” ou “elbow method” é uma técnica usada para encontrar a quantidade ideal de clusters K. Este método testa a variância dos dados em relação ao número de clusters. O valor ideal de K é aquele que tem um menor Within Sum of Squares (WSS) e ao mesmo tempo o menor número de clusters.

Através do código abaixo, fizemos a análise dos dados na tentativa de encontrar a melhor quantidade de cluster. Aplicamos a análise simulando até 30 clusters (Figura 68 - Análise do vector através da curva de cotovelo):

```
#identificando a qtde de clusters ideal através do método do cotovelo
from sklearn.cluster import KMeans
wcss = []
for i in range(1,30):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=0)
    kmeans = kmeans.fit(tfidf_vector)
    print(i, kmeans.inertia_)
    wcss.append(kmeans.inertia_)
```

```
1 61441.66552849984
2 60806.532952498506
3 60431.238896361814
4 59944.30767654894
5 59652.37097554814
6 59380.43240767079
7 59094.72123381449
8 58950.15658590595
9 58720.45924338682
10 58339.5664665085
11 58242.93504074296
12 57962.92630676399
13 57762.44665161848
14 57806.47944446535
15 57536.10785050134
16 57479.5578234724
17 57239.38855160987
18 57104.20359519952
19 56984.88761792124
20 56822.13321127679
21 56826.44241830401
22 56601.995626313445
23 56586.0315854936
24 56479.7648944236
25 56310.59096796363
26 56146.823705931296
27 56044.83952361166
28 56024.91751211195
29 55931.42840660707
```

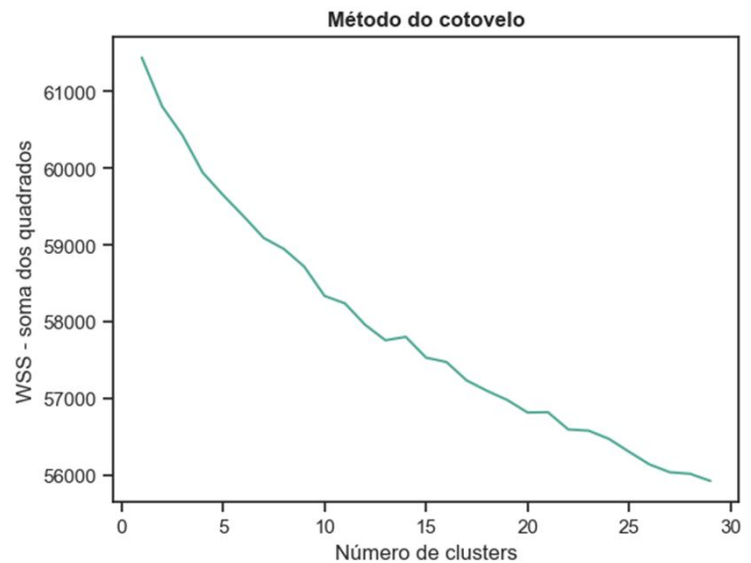


Figura 68 - Análise do vector através da curva de cotovelo

Não foi possível determinar a quantidade de clusters (K) através do “método de cotovelo” pois não houve um ponto ideal de inflexão da curva, apontando a estabilidade na quantidade de clusters.

5.3.1.4.2 Silhouette

No método de “Silhouette” é analisado um coeficiente resultante do cálculo da distância entre os centroides, levando em consideração o agrupamento dos dados que os cerca. O coeficiente é gerado entre o intervalo de -1 a +1. Quanto mais próximo a “+1” representa maior distância entre os clusters e melhor resultado esperado; quando for “0” significa que estão muito próximos do ponto de decisão entre os clusters, ou seja, impossibilita a decisão a qual cluster pertence; e quando os valores forem negativos isso significa que possivelmente os dados estão no cluster errado. Também deve ser levado em consideração na análise o valor médio do coeficiente, sendo que, no cenário ideal, o coeficiente de silhueta de cada cluster deve ser maior que o valor médio do mesmo e as espessuras de cada um no gráfico devem ser semelhantes entre si.

Após a aplicação do método de “Silhouette”, identificamos que também não foi possível a determinação da quantidade de clusters (K) através desse método pois todos os coeficientes ficaram próximos de zero (0). Vide código e gráfico abaixo (Figura 69 - Cálculo e representação do coeficiente de Silhouette):

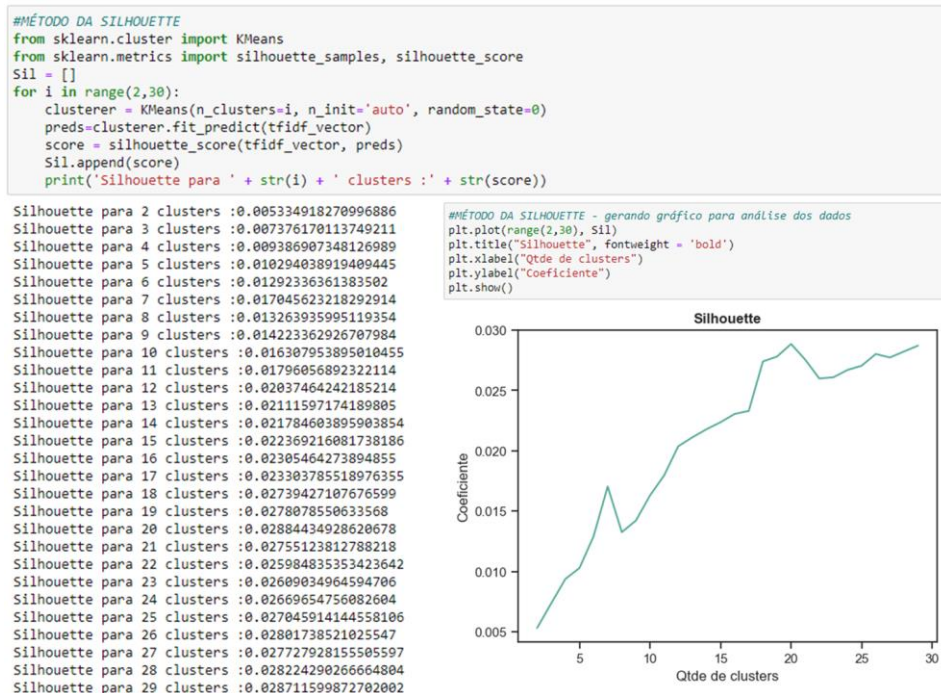


Figura 69 - Cálculo e representação do coeficiente de Silhouette

5.3.1.4.3 Aplicação do K-Means

Devido a impossibilidade de determinarmos a quantidade ideal de clusters, adotaremos a quantidade padrão do “K-Means” de dez (10) clusters (K). Essa é a quantidade de clusters limite para ser processado com os recursos atuais de máquina (Core i5, vPro, 8th Gen, 8 GB Ram). O parâmetro Random_state foi parametrizado com valor “0 (zero)” para que a reprodução do modelo resulte no mesmo resultado caso seja processado novamente. O resultado do treino do modelo será registrado em uma nova coluna no dsCAC_SemOutliers (“Cluster”), e os valores serão os respectivos números de cada “cluster” (valores de 0 - 9). Dessa forma, passamos a identificar a qual cluster cada atendimento CAC pertence (Figura 70 - Treinamento e alocação em clusters):

```
#Treinamento do modelo
from sklearn.cluster import KMeans
num_clusters = 10
kmeans = KMeans(n_clusters=num_clusters, init='k-means++',
                 n_init=10, max_iter = 300, random_state=0)
kmeans = kmeans.fit(tfidf_vector)
centroides = kmeans.cluster_centers_
labels = kmeans.labels_
dsCAC_SemOutliers['Cluster'] = kmeans.fit_predict(tfidf_vector)

print(dsCAC_SemOutliers['Cluster'].value_counts())
```

| Cluster | |
|---------|-------|
| 0 | 30045 |
| 3 | 11862 |
| 6 | 6249 |
| 4 | 5692 |
| 2 | 3176 |
| 5 | 1866 |
| 7 | 1724 |
| 9 | 907 |
| 8 | 519 |
| 1 | 372 |

Name: count, dtype: int64

Figura 70 - Treinamento e alocação em clusters

Representação gráfica da alocação dos termos por cluster (Figura 71 - Resultado da clusterização):

```
#Demonstração gráfica dos clusters
fig, ax = plt.subplots(figsize=(8,5))
sns.countplot(data=dsCAC_SemOutliers,x="Cluster",palette='rocket')
plt.title('Alocação dos atendimentos por cluster',fontsize=16, fontweight='bold')
ax.set_xlabel('Cluster',fontsize=16, fontweight='bold')
ax.set_ylabel('Qtde. atendimentos',fontsize=12, fontweight='bold')
plt.grid(True, axis='y')
plt.show()
```

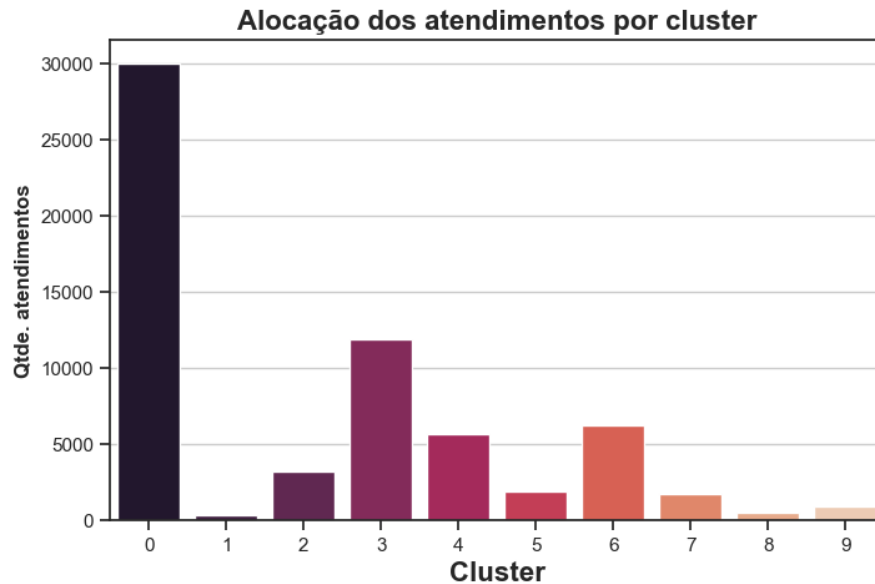


Figura 71 - Resultado da clusterização

Para uma análise imediata de uma amostra dos clusters, geramos uma lista com as 10 principais palavras de cada cluster (Figura 72 - Palavras mais frequentes por cluster):

```
print('Palavras mais frequentes por cluster:')
order_centroids = centroides.argsort()[::-1]
terms = count_vect.get_feature_names_out()
for i in range(10):
    print('Cluster --> {}'.format(i))
    for ind in order_centroids[i,:10]:
        print(' %s' % terms[ind],end='')
    print()
```

Palavras mais frequentes por cluster:

Cluster --> 0:
solicitar protocolo sobre entrar comentario gostar boa questionar informacao tarde

Cluster --> 1:
reset senhar bateria club acesso email alteracao solicitar realizar desejar

Cluster --> 2:
radio codigo solicitar entrar desejar desbloqueio nao informar manual bloquear

Cluster --> 3:
informar nao entrar conseguir fazer pois precisar reclamacao chassi solicitar

Cluster --> 4:
desejar saber sobre banco comprar informacoes boleto pecas informar falar

Cluster --> 5:
revisao agendar questionar periodo realizar desejar sobre saber fazer informar

Cluster --> 6:
consultor acompanhamento pecar entrada nao envolver data mail reclamacao reparo

Cluster --> 7:
sg relacionar assunto contar conta reconsideracao res erro negar re

Cluster --> 8:
atualizacao dar cadastral plantao sh shrs cadastro gs horas informacoes

Cluster --> 9:
fechamento protocolo envio gerar mail reenvio res solicitacao enc re

Figura 72 - Palavras mais frequentes por cluster

Outra forma de visualizarmos os clusters é através da plotagem de um gráfico de barras, mostrando o TF-IDF de cada palavra no seu respectivo cluster (Figura 74 - Plotagem dos 10 clusters e Figura 75 - Representação gráfica do TF-IDF do cluster e as 10 principais palavras). Para isso, geramos um dataset com o número do cluster, palavra e TF-IDF (Figura 73 - Preparação do dataset (cluster, palavra e TF-IDF)):

```
#preparação do dataset
bow_transformer = count_vect.fit(dsCAC_SemOutliers['Descricao'])

print('Palavras mais frequentes por cluster:')
order_centroids = centroides.argsort()[::-1]
lista_terms=[]

for i in range(0,10):
    print('\n Cluster {}: '.format(i))
    for x in order_centroids[i,:10]:
        cluster = i
        term=feature_names[x]
        valor = tfidf_transformer.idf_[bow_transformer.vocabulary_[term]]
        print(term, ': ', valor)
        lista_terms.append([i, term, valor])
dsPalavras = pd.DataFrame(lista_terms, columns=['Cluster','Palavra', 'TF-IDF'])
```

Figura 73 - Preparação do dataset (cluster, palavra e TF-IDF)

```
#plotagem do gráfico dos 10 clustes com suas palavras e respectivos TF-IDF
plt.rcParams["figure.figsize"] = [15.00, 12.00]
plt.rcParams["figure.autolayout"] = True
num_linhas = 5
num_colunas = 2
fig, ax = plt.subplots(num_linhas,num_colunas)

linha = 0
coluna = 0

for i in range(0,10):
    plt.title('Palavras mais frequentes no Cluster {}'.format(i), fontsize=14, fontweight='bold')
    plt.xlabel('TF-IDF', fontsize=12, fontweight='bold')
    plt.ylabel('Palavra', fontsize=12, fontweight='bold')
    sns.barplot(data=dsPalavras[dsPalavras['Cluster']==i], x='TF-IDF',y='Palavra',
                orient='h', palette='YlGnBu',ax=ax[linha][coluna])
    coluna += 1
    if coluna == num_colunas:
        linha += 1
        coluna = 0
plt.show()
```

Figura 74 - Plotagem dos 10 clusters

Código e representação gráfica dos 10 clusters, 10 principais palavras de cada cluster e TF-IDF (Figura 74 - Plotagem dos 10 clusters):

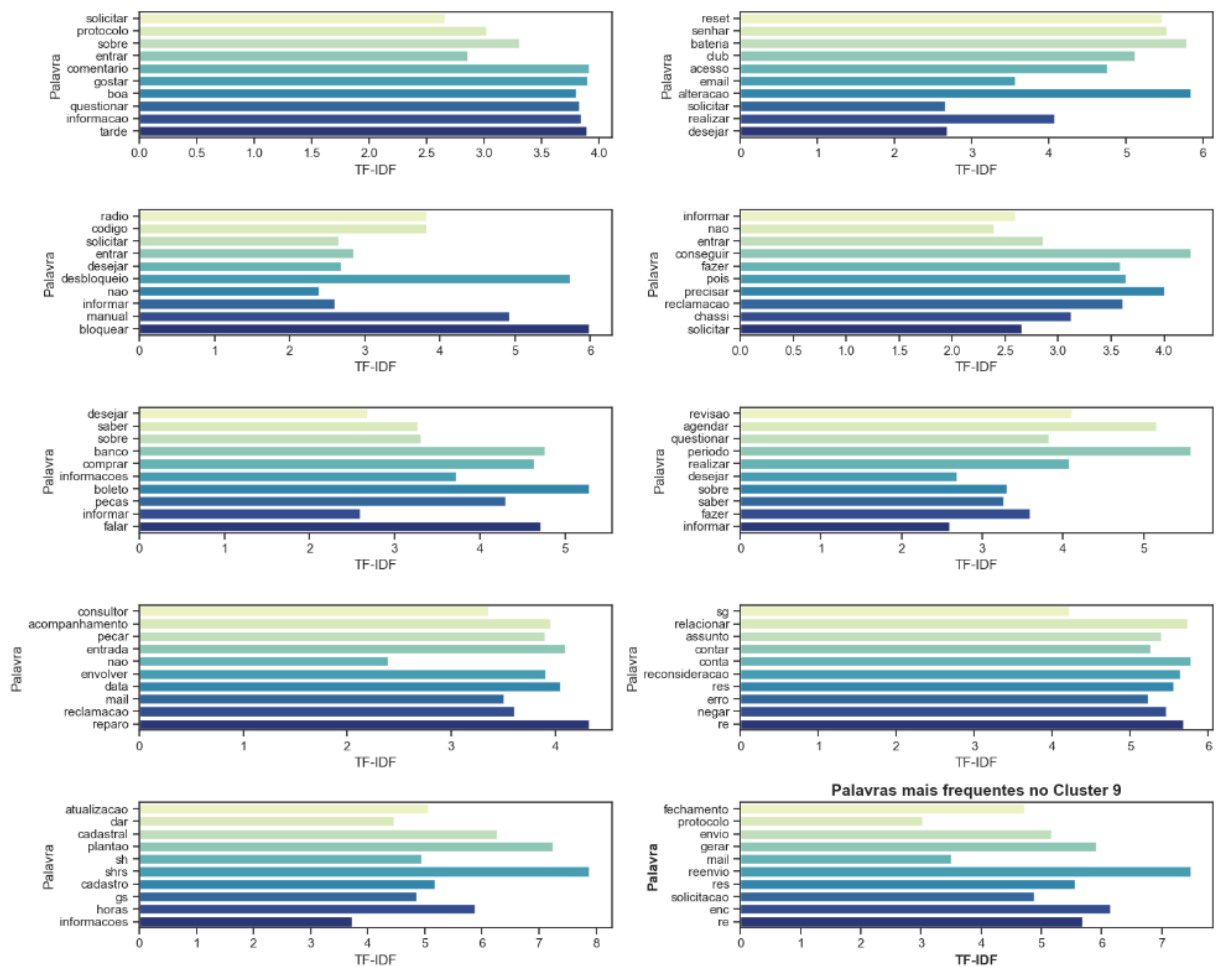


Figura 75 - Representação gráfica do TF-IDF do cluster e as 10 principais palavras

Os clusters “0”, “3” e “6” são os que possuem mais palavras, sendo o cluster “0” o que possui 50,0% de todas as palavras clusterizadas (Figura 76 - Percentual de palavras por cluster):

```
dsPercCluster = dsCAC_SemOutliers['Cluster'].value_counts().reset_index()

dsPercCluster['Percentual'] = dsPercCluster['count']/dsCAC_SemOutliers.shape[0]*100

dsPercCluster = dsPercCluster.sort_values(by=['Percentual'], ascending=False)

dsPercCluster
```

| | Cluster | count | Percentual |
|---|---------|-------|------------|
| 0 | 0 | 30045 | 48.139781 |
| 1 | 3 | 11862 | 19.005960 |
| 2 | 6 | 6249 | 10.012498 |
| 3 | 4 | 5692 | 9.120041 |
| 4 | 2 | 3176 | 5.088765 |
| 5 | 5 | 1866 | 2.989810 |
| 6 | 7 | 1724 | 2.762289 |
| 7 | 9 | 907 | 1.453246 |
| 8 | 8 | 519 | 0.831571 |
| 9 | 1 | 372 | 0.596039 |

Figura 76 - Percentual de palavras por cluster

Abaixo, apresentamos uma amostragem dos 5 primeiros e 5 últimos registros do cluster "0" (Figura 77 - Amostragem do cluster "0"):

```
dsCAC_SemOutliers[dsCAC_SemOutliers['Cluster'] ==0]['Descricao']
```

```
2                acesso tcas
3    solangir situacao solange entrar solicitar inf...
4    selma protocolo selma entrar querer saber real...
5    edson situacao edson entrar solicitar infomaco...
7    comentario solicito email cadastrar gentileza ...
...
63758                processo pagamento
63761                ativacao servico
63764                processo pagamento
63766    r nepomuceno questionar sobre ativacao servico
63768    processo pagamento boa tarde sg obrigado descr...
Name: Descricao, Length: 30045, dtype: object
```

Figura 77 - Amostragem do cluster "0"

6. Interpretação dos Resultados

A central de atendimento ao cliente, pela sua pluralidade dos meios de contato (telefone, e-mail, chat, site, aplicativos específicos e etc.) entre o Cliente e a empresa prestadora do produto e/ou serviço é o principal meio de contato entre as partes mencionadas.

Os contatos recebidos possuem propósitos variados, navegando desde a solicitação de atendimento à veículo parado na rodovia, necessitando de suporte técnico e guincho, até empresas e pessoas oferecendo serviços para a fabricante do produto/serviço, solicitando doações, enviando curriculum e etc.

Identificamos que durante o atendimento, o atendente, por necessidade de acompanhamento do contato, inclui na descrição do atendimento todas as informações que possam facilitar a solução do problema: pessoas de contato (nomes), e-mails, telefones, nome da empresa, descrição do problema, etc. Para potencializar a análise dos dados, é importante padronizar o máximo possível a captação dos dados, classificando-as e agrupando-as em categorias específicas (registros/campos específicos no sistema da CAC).

O trabalho conjunto entre o analista de negócio (pessoa com profundo conhecimento do produto e serviço) e o cientista de dados é fundamental para o êxito do trabalho.

As etapas de modelagem e preparação dos dados para machine learning é fundamental e crucial para o resultado da aplicação dos algoritmos e deve ser investido tempo e atenção nessa etapa. O envolvimento de diferentes áreas da empresa se torna importante nessa fase (Ex.: área de negócios (vendas e pós-venda), TI, jurídica, engenharia, etc.) e as etapas abaixo devem ser cumpridas:

- 1) Entendimento do domínio do problema
- 2) Montagem da base de dados
- 3) Preparação da base de dados
- 4) Redução da dimensionalidade e seleção de atributos

Durante o desenvolvimento desse estudo, foi necessário retroceder algumas vezes para as etapas 2) e 3) e solicitar apoio contínuo do analista de negócios e área de TI. Foi necessário também o refinamento de alguns parâmetros durante o processamento dos algoritmos e bibliotecas de machine learning (etapa 4). O ajuste do “stopwords”, incluindo palavras

novas para serem desconsideradas e consideradas também contribuiu para o melhor resultado do trabalho.

Como exemplo do refinamento necessário nas bibliotecas nltk e spacy, a palavra “peça” (no contexto estudado significa uma peça do veículo), foi transformada em “peca” (padronização da palavra através da biblioteca “unidecode”) e interpretada como sendo do verbo “pecar” pela biblioteca nltk.

Existem diversas bibliotecas disponíveis para tratamento, processamento de dados e aplicação de machine learning. Investir tempo e atenção no estudo do processo e ferramentas que serão utilizadas é muito importante e podem levar a resultados distintos.

Os recursos de processamento de máquina causaram algumas restrições ao trabalho aqui desenvolvido. Exemplo foi a tentativa de redução da dimensionalidade (5.3.1.3 Análise e aplicação da redução da dimensionalidade).

7. Apresentação dos Resultados

O objetivo desse estudo, como mencionado no item **1.3. Objetivos**, é analisarmos e identificarmos padrões no que é relatado durante o atendimento via CAC (sentimentos), para direcionar ações de melhorias nas áreas vendas e pós-venda.

Analisamos 73.029 registros/contatos com a Central de Atendimento ao Cliente (CAC), em sua maioria por Clientes da marca estudada, de Abr/2019 à Jul/2023. Desses, 5.127 contatos foram para solicitar informações corporativas, podendo ou não ter sido feito por Cliente da marca.

Após a limpeza, padronização e filtragem dos dados (mantivemos no dataset somente os segmentos de caminhões e ônibus), o dataset foi reduzido para 70.410 registros.

O CAC atende as 27 unidades da federação (Estados) e a proporção percentual do volume de atendimento via CAC é condizente com a frota circulante nacional e a quantidade de RVAT (Rede de Vendas e Assistência Técnica) de cada região. A tabela abaixo mostra de forma clara essa proporção onde a UF de SP é a mais representativa (Figura 78 - Percentual de RVAT, Frota circulante e atendimento CAC por UF):

| UF | Rede de vendas e assistência técnica | | | Frota circulante | | | Atendimento via CAC | | |
|----|--------------------------------------|------------|------------|------------------|------------|------------|---------------------|------------|------------|
| | Qtde. | Percentual | Per. Acum. | Qtde. | Percentual | Per. Acum. | Qtde. | Percentual | Per. Acum. |
| SP | 48 | 21,7% | 21,7% | 404.459 | 23,1% | 23,1% | 5.571 | 22,9% | 22,9% |
| MG | 29 | 13,1% | 34,8% | 248.758 | 14,2% | 37,4% | 4.244 | 17,5% | 40,4% |
| PR | 21 | 9,5% | 44,3% | 152.340 | 8,7% | 46,1% | 1.840 | 7,6% | 48,0% |
| RS | 20 | 9,0% | 53,4% | 149.306 | 8,5% | 54,6% | 1.185 | 4,9% | 52,8% |
| SC | 16 | 7,2% | 60,6% | 81.921 | 4,7% | 59,3% | 1.324 | 5,4% | 58,3% |
| RJ | 12 | 5,4% | 66,1% | 100.607 | 5,8% | 65,0% | 1.046 | 4,3% | 62,6% |
| MT | 10 | 4,5% | 70,6% | 51.548 | 2,9% | 68,0% | 1.025 | 4,2% | 66,8% |
| GO | 9 | 4,1% | 74,7% | 65.869 | 3,8% | 71,8% | 1.247 | 5,1% | 72,0% |
| BA | 8 | 3,6% | 78,3% | 76.362 | 4,4% | 76,1% | 1.821 | 7,5% | 79,4% |
| PA | 6 | 2,7% | 81,0% | 40.526 | 2,3% | 78,4% | 644 | 2,7% | 82,1% |
| ES | 5 | 2,3% | 83,3% | 58.596 | 3,4% | 81,8% | 496 | 2,0% | 84,1% |
| PE | 5 | 2,3% | 85,5% | 54.498 | 3,1% | 84,9% | 406 | 1,7% | 85,8% |
| PI | 4 | 1,8% | 87,3% | 20.251 | 1,2% | 86,1% | 257 | 1,1% | 86,9% |
| CE | 4 | 1,8% | 89,1% | 45.946 | 2,6% | 88,7% | 411 | 1,7% | 88,6% |
| MS | 3 | 1,4% | 90,5% | 33.176 | 1,9% | 90,6% | 424 | 1,7% | 90,3% |
| RO | 3 | 1,4% | 91,9% | 19.479 | 1,1% | 91,7% | 355 | 1,5% | 91,8% |
| PB | 3 | 1,4% | 93,2% | 18.213 | 1,0% | 92,7% | 183 | 0,8% | 92,5% |
| RN | 2 | 0,9% | 94,1% | 19.226 | 1,1% | 93,8% | 166 | 0,7% | 93,2% |
| SE | 2 | 0,9% | 95,0% | 16.346 | 0,9% | 94,8% | 161 | 0,7% | 93,9% |
| TO | 2 | 0,9% | 95,9% | 11.262 | 0,6% | 95,4% | 335 | 1,4% | 95,2% |
| AL | 2 | 0,9% | 96,8% | 15.589 | 0,9% | 96,3% | 182 | 0,7% | 96,0% |
| MA | 2 | 0,9% | 97,7% | 23.403 | 1,3% | 97,6% | 406 | 1,7% | 97,7% |
| RR | 1 | 0,5% | 98,2% | 2.743 | 0,2% | 97,8% | 3 | 0,0% | 97,7% |
| DF | 1 | 0,5% | 98,6% | 18.715 | 1,1% | 98,9% | 337 | 1,4% | 99,1% |
| AP | 1 | 0,5% | 99,1% | 1.891 | 0,1% | 99,0% | 21 | 0,1% | 99,1% |
| AM | 1 | 0,5% | 99,5% | 12.754 | 0,7% | 99,7% | 139 | 0,6% | 99,7% |
| AC | 1 | 0,5% | 100,0% | 5.011 | 0,3% | 100,0% | 68 | 0,3% | 100,0% |
| | | | | 1.748.795 | | | 24.297 | | |

Figura 78 - Percentual de RVAT, Frota circulante e atendimento CAC por UF

O que chama a atenção é que o estado de “SC”, embora possua uma frota menor do que o estado do “RJ”, ela possui mais RVAT e atendimentos via CAC. A explicação é relacionada

ao tamanho dos estados, “SC” com 95.346 km² e “RJ” com 43.696 km², demandando mais RVAT para cobrir os atendimentos. Outro fator relevante é que a marca estudada possui uma grande participação de vendas de ônibus no estado do “RJ” e os Clientes frotistas de ônibus, em sua maioria, possuem oficina própria para manutenção e atendimento de veículo parado nas ruas e avenidas, não demandando dessa forma, apoio da CAC do fabricante.

Nota: dos 70.410 atendimentos analisados, 24.297 foi possível atribuir a qual RVAT pertencia pois não é mandatório e/ou sempre necessário a atribuição do atendimento à um RVAT.

O gráfico de calor abaixo mostra a concentração dos atendimentos via CAC (Figura 79 – Mapa de calor: Concentração dos atendimentos via CAC):

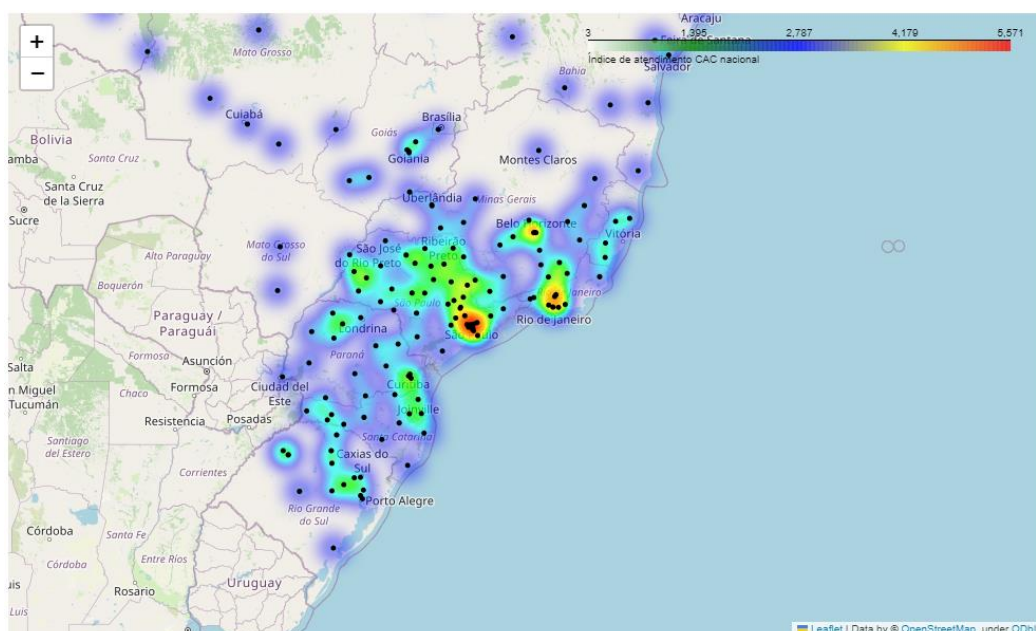


Figura 79 – Mapa de calor: Concentração dos atendimentos via CAC

Uma outra análise nos dados mostra que os principais motivos do contato do Cliente ao CAC estão relacionados ao “Produto” (questões técnicas), atendimento e qualidade da RVAT e Pós Venda da marca estudada. Esses são indicadores das áreas que necessitam de atenção e ações mais contundentes (Figura 80 - Classificação do atendimento via CAC):

| Classificação do atendimento | Qtde. | Percentual | Per. Acum. |
|--------------------------------------|--------|------------|------------|
| Reclamação/Questionamento do produto | 17.959 | 25,5% | 25,5% |
| Questionamento ao Concessionário | 14.089 | 20,0% | 45,5% |
| Qualidade Pós Venda | 13.381 | 19,0% | 64,5% |
| Questionamento do Serviço | 11.281 | 16,0% | 80,5% |
| Informação corporativa | 5.127 | 7,3% | 87,8% |
| Transferência do atendimento | 3.616 | 5,1% | 93,0% |
| Questionamento de vendas | 1.933 | 2,7% | 95,7% |
| Agendando reparo | 1.615 | 2,3% | 98,0% |
| Qualidade Vendas | 966 | 1,4% | 99,4% |
| Agendamento serviço | 207 | 0,3% | 99,7% |
| Suporte assistência técnica | 171 | 0,2% | 99,9% |
| Agendamento de recall | 40 | 0,1% | 100,0% |
| Agendamento vendas | 15 | 0,0% | 100,0% |
| Serviço de contrato de manutenção | 6 | 0,0% | 100,0% |
| Reparo acidente | 4 | 0,0% | 100,0% |
| 70.410 | | | |

Figura 80 - Classificação do atendimento via CAC

O processamento das palavras captadas durante os atendimentos via CAC, aplicando as metodologias Bag of Words, TF-IDF e K-Kmeans, captadas durante os atendimentos via CAC resultou no agrupamento das palavras em 10 clusters (pacotes). Os clusters 0, 3 e 6 são os que representam o maior percentual dos agrupamentos (aprox. 80,0%). Veja gráfico abaixo (Figura 81 - Alocação dos atendimentos por cluster):

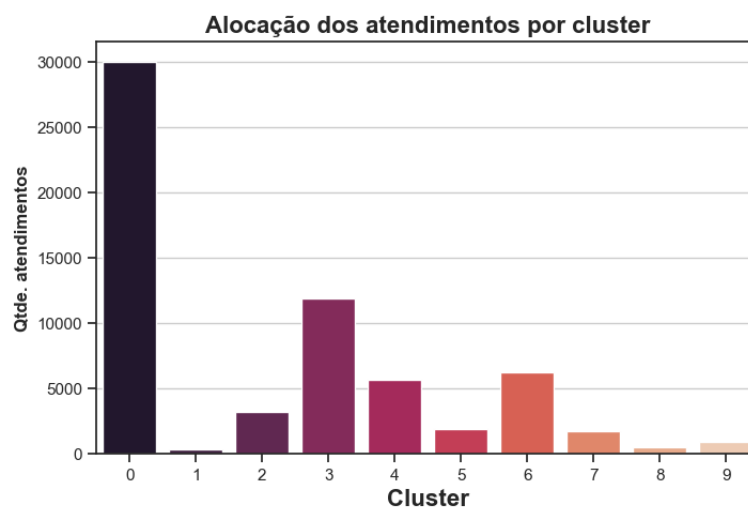


Figura 81 - Alocação dos atendimentos por cluster

Pela análise do gráfico das 10 principais palavras por cluster (Figura 82 - 10 principais palavras constantes nos três maiores clusters), notamos que os principais motivos de contato ao CAC são referentes a necessidade do Cliente em resolver o problema técnico do veículo e colocá-lo novamente em operação: reparo, conseguir, data, precisar, peça, solicitar, informar e não.

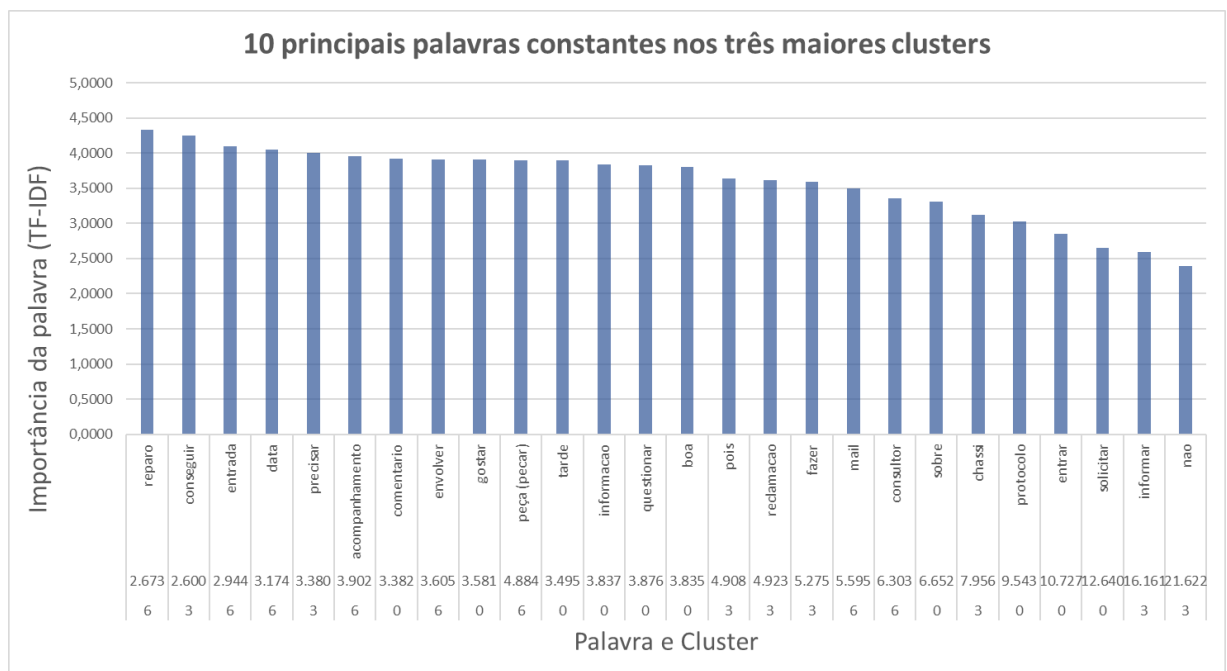


Figura 82 - 10 principais palavras constantes nos três maiores clusters

Importante ressaltar que não é a quantidade de vezes que uma palavra aparece na análise (nuvem de palavras) e sim a importância da palavra medida através do TF-IDF (Figura 83 - Cálculo do TF-IDF).

$$TFIDF = TF \times IDF$$

ou

$$TFIDF = \frac{\text{Nº DE VEZES QUE UMA PALAVRA APARECE EM UM DOCUMENTO}}{\text{Nº DE PALAVRAS DO DOCUMENTO}} \times \log \left(\frac{\text{TOTAL DE DOCUMENTOS}}{\text{Nº DE DOCUMENTOS COM O RESPECTIVO TERMO}} \right)$$

Figura 83 - Cálculo do TF-IDF

Analisando pela importância da palavra, concluímos que “shrs” é o termo mais importante na análise da base de dados da CAC. “shrs” significa o serviço 24 horas, que atende problemas técnicos nas rodovias brasileiras em todo o território nacional. Outro termo referente ao serviço 24 horas é “plantao”.

Outras palavras que chamaram a atenção e foram confirmadas pelo analista de negócios são:

Peça (pecar): referente a reclamação constante de falta de peças de reposição na RVAT para consertar o veículo

Bateria: descarregamento da bateria pela utilização não balanceada de equipamentos eletrônicos enquanto o veículo está com o motor desligado

Desbloqueio: referente ao bloqueio do rádio após o desligamento completo da bateria, sendo necessário a inclusão do código de desbloqueio do rádio após a reconexão da bateria (Figura 84 - Palavras com maior importância (TF-IDF):

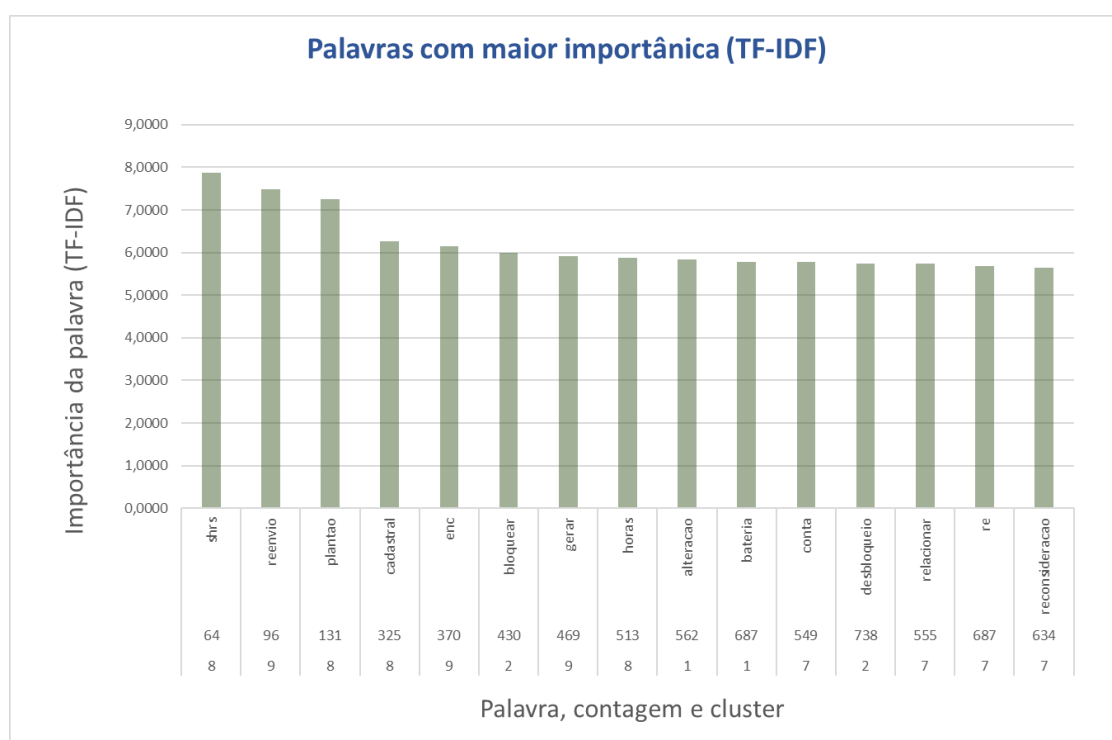


Figura 84 - Palavras com maior importância (TF-IDF)

Analizando a concordância (os contextos que a palavra aparece no texto) e a similaridade (quais palavras mais aparecem no mesmo contexto de uma determinada palavra) da palavra “desbloqueio”, vemos que ela está diretamente relacionada a “radio”, “código” e “informações”, sendo assim, um direcionamento para a área de Pós Venda simplificar a obtenção do código do rádio pelo Cliente, sem ser necessário contatar a CAC (a CAC não fornece o código do rádio e sim orienta o Cliente a procurar uma RVAT). Veja respectivamente as Figura 85 - Análise da concordância da palavra "desbloqueio" e Figura 86 - Análise da similaridade da palavra "desbloqueio":

```

Displaying 25 of 738 matches:
tratar solicitar informacoes codigo desbloqueio radio nao manual informei questao
o poder orienta lo fornecer codigo desbloqueio informei endereco numero saber in
ntrar solicitar informacoes codigo desbloqueio radio expectativa informei questao
o poder orienta lo fornecer codigo desbloqueio forneceu numero inga sao miguel d
ntrar solicitar informacoes codigo desbloqueio radio expectativa informei questao
tratar solicitar informacoes codigo desbloqueio radio informei codigo radio manua
o poder orienta lo fornecer codigo desbloqueio informei numero rodobens salvador
entrar visor radio informar codigo desbloqueio nao conseguir colocar regivaldo e
tratar solicitar informacoes codigo desbloqueio radio nao codigo manual informei
o poder orienta lo fornecer codigo desbloqueio informei numero peres diesel avar
olicitar informacoes sobre codigo desbloqueio radio seguir arquivo detalhes agu
ntrar solicitar informacoes codigo desbloqueio radio expectativa informei questao
o poder orienta lo fornecer codigo desbloqueio forneceu numero cardiesel belo ho
ntrar solicitar informacoes codigo desbloqueio radio expectativa informei questao
ntrar solicitar informacoes codigo desbloqueio radio expectativa informei questao
oculo possuir atego gostar codigo desbloqueio radio manual operacao joao vitor
ntrar solicitar informacoes codigo desbloqueio radio expectativa informei questao
tratar solicitar informacoes codigo desbloqueio radio nao manual operacao informe
o poder orienta lo fornecer codigo desbloqueio informei concessionarias cardiese
tratar solicitar informacoes codigo desbloqueio radio nao manual operacao informe
o poder orienta lo fornecer codigo desbloqueio informei endereco numero agricolt
tratar solicitar informacoes codigo desbloqueio radio manual informei questao seg
o poder orienta lo fornecer codigo desbloqueio informei pirasa limeira sp fone p
acao catrieli entrar pedir codigo desbloqueio radio expectativa informei mesma
ntrar solicitar informacoes codigo desbloqueio radio expectativa informei questao

```

Figura 85 - Análise da concordância da palavra "desbloqueio"

```

radio codigo desbloquear informar informacao manual cor informacoes
peca pecar entrar nao pois reparo revisao orientei por auxilio
novamente inserir

```

Figura 86 - Análise da similaridade da palavra "desbloqueio"






Como conclusão geral, é importante ressaltar que esse projeto executou o algoritmo de aprendizado de máquina não supervisionado, sendo o resultado interpretado de forma subjetiva pela falta de critérios de agrupamento. A análise minuciosa do resultado pelo analista de negócios é fundamental para a implementação prática dos resultados obtidos.






The machine learning CANVAS

Designed for: PUC Minas

Designed by: Alessandro Zabotto

Date: 08/10/2023 Iteration: -

| PREDICTION TASK  | DECISIONS  | VALUE PROPOSITION  | DATA COLLECTION  | DATA SOURCES  |
|--|--|--|--|---|
| <p><i>Analisar os dados captados durante o atendimento ao cliente através da “Central de Atendimento ao Cliente” de uma empresa do setor automobilístico.</i></p> <p><i>O resultado esperado é a análise de sentimento e clusterização do motivo do contato a CAC, analisando o texto captado pelo atendente (corpus).</i></p> | <p><i>Utilização do processamento de linguagem natural para clusterização e classificação das palavras/termos mais comuns, canalizando os esforços da CAC e demais áreas da empresa.</i></p> | <p><i>Melhorar o atendimento ao cliente através da análise, implementação e aplicação de machine learning nos textos captado pelos atendentes da CAC Identificar de forma mais assertiva o correto direcionamento do atendimento (Ex.: assunto comercial, técnico, institucional e etc.)</i></p> | <p><i>Os dados foram obtidos da base de dados do sistema da CAC e dados públicos do setor de transporte.</i></p> | <p><i>A extração de novos dados deve ser programada no sistema que gerencia a CAC e disponibilizada em servidor de dados específico para serem processados conforme programação feita pelo cientista de dados (TI). A entidade de classe do setor automobilístico não disponibiliza recursos para obtenção dos dados através de API/Web-service. Dessa forma, os dados devem ser baixados direto no site e convertidos para o formato .csv para ser processado.</i></p> |

| | | | | |
|---|--|--|--|---|
| <p>IMPACT SIMULATION </p> <p><i>Os modelos podem ser implementados na íntegra, não necessitam de alto poder de processamento e não existe impedimentos legais ou técnicos.</i></p> | <p>MAKING PREDICTIONS </p> <p><i>Novos processamentos devem ser feitos uma vez ao mês durante o primeiro ano, com potencial para passarem a ser processados uma vez a cada seis meses tão logo as análises se estabilizem.</i></p> | | <p>BUILDING MODELS </p> <p><i>Os modelos utilizados serão: Bag Of Words, TF-IDF e K-Means. Os modelos serão reavaliados semestralmente.</i></p> | <p>FEATURES </p> <p><i>Novos produtos ou serviços podem impactar a clusterização dos dados, sendo necessário reavaliar e/ou parametrizar os modelos de aprendizagem.</i></p> |
| | <p>MONITORING </p> <p><i>Acompanhar a mudança de posicionamento dos termos/palavras e/ou novos termos mais frequentes devido a entrada de novos produtos, serviços, ferramentas, concorrentes e comportamento dos clientes.</i></p> | | | |

8. Links

Segue link para o vídeo de apresentação do trabalho de conclusão do curso:

Link para o vídeo: <https://youtu.be/qmrb014HDQQ>

Segue link da dissertação do trabalho de conclusão do curso e demais documentos que foram base para esse trabalho:

Link para o repositório: https://github.com/asaboto/PucMinas_TCC

Referências

Abaixo estão os links e referências aos livros, artigos, publicações, sites técnicos mantido por terceiros e sites oficiais das ferramentas, bibliotecas e recursos que serviram como fonte de estudo para o desenvolvimento desse estudo:

- ChengXiang Zhai and Sean Massung. 2016. Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining. Association for Computing Machinery and Morgan & Claypool.
- Wes McKinney and the Pandas Development Team, pandas: powerful Python data analysis toolkit Release 1.4.4 - Aug 31, 2022 (<https://pandas.pydata.org/pandas-docs/version/1.4/pandas.pdf>)
- Matplotlib (<https://matplotlib.org/>), © Copyright 2002–2012 John Hunter, Darren Dale, Eric Firing, Michael Droettboom and the Matplotlib development team; 2012–2023 The Matplotlib development team. Created using [Sphinx](#) 5.3.0. Built from v3.7.2-2-g4915f574b8.
- Seaborn (<https://seaborn.pydata.org/>), © Copyright 2012–2022, [Michael Waskom](#). Created using [Sphinx](#) and the [PyData Theme](#).
- Acervolima (<https://acervolima.com/>), © 2022 Acervo Lima, Some rights reserved
- Python 3.11.4 documentation (<https://docs.python.org/3/>), © Copyright 2001–2023, Python Software Foundation
- Scikit (<https://scikit-learn.org/>), © 2007 - 2023, scikit-learn developers (BSD License)

Apêndice

Programação/Scripts

O script foi disponibilizado na íntegra no Github.

Gráficos e tabelas

Todos os gráficos e tabelas foram inseridos no texto principal desse estudo.