hefEStos
Rede Gazeta

inteli
instituto
de tecnologia
e liderança

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
08/08/2022	Marcos, Priscila, Maria Luísa, Matheus e Henrique	1.1	Criação do documento Edição do tópico 4.1.1 Edição do tópico 4.1.4 Edição do tópico 4.1.5
09/08/2022	Pedro Priscila Marcos	1.2	Edição do tópico 4.1.2 Edição do tópico 4.1.3 Inserção de dados nos subtópicos do tópico 4
10/08/2022	Maria Luísa Marcos Pedro	1.3	Revisão e conclusão dos tópicos do artefato 1 (4.1.1, 4.1.2, 4.1.3, 4.1.4 e 4.1.5)
15/08/2022	Maria Luisa Pedro Rafael Henrique Marcos Matheus	2.1	Fazer 4.2 - análise de dados
17/08/2022	Pedro Priscila Matheus	2.2	Persona Jornada de usuário
20/08/2022	Priscila	2.3	Formatação da documentação 4.1.2 - Descrição da matriz SWOT
24/08/2022	Maria Luisa	2.4	Revisão dos tópicos dos artefatos da sprint 1
25/08/2022	Maria Luisa Henrique Matheus	2.5	Formatação + passar para outro documento Realização do tópico 4.3

26/08/2022	Maria Luisa Priscila Falcão	2.6	Revisão de formatação, ortografia e conteúdo Tópicos 4.3.2 e 4.3.3
29/08/2022	Priscila Falcão	3.1	Texto de personas; Texto de jornada de usuário; Texto de matriz de risco;
30/08/2022	Maria Luisa Priscila Falcão	3.2	Revisão análise SWOT CRISP-DM
07/09/2022	Priscila Falcão	3.3	CRISP-DM Formatação código

Sumário

1. Introdução

A **Rede Gazeta de Comunicações**, ou **Rede Gazeta**, é um conjunto de mídia brasileiro localizado no estado do Espírito Santo. Possuindo mais de 500 funcionários, a Rede Gazeta é o maior grupo de comunicação do Espírito Santo, a empresa foi fundada em 1928, com o jornal A Gazeta, porém apenas no ano de 1976 a TV Gazeta surgiu, aproveitando o grande crescimento dos meios de comunicação em massa. Atualmente, eles contam com a presença de 16 veículos de comunicação, abrangendo a TV, rádio e internet. Contudo, nos últimos anos, viu-se necessária a criação de um meio para melhorar a média de audiência dessa emissora

Nos últimos anos essa emissora cresceu bastante pelas suas produções próprias, porém com a grande competitividade das plataformas de streaming e das redes sociais, é de plena importância a constante pesquisa e análise de dados para suas próximas empreitadas. Por isso, o grupo hefEStos realizou a criação de um software que através da análise de dados e a partir da inteligência artificial realiza um modelo preditivo que ajudará na previsão da audiência de novos programas.

2. Objetivos e Justificativa

2.1. Objetivos

A Rede Gazeta procurou o Inteli para fazer esse projeto com os alunos, como o grupo hefEStos, para poder analisar e prever quais programas de TV são potenciais produtos para um investimento e expectativa maior.A empresa propôs a construção de um software que contará com machine learning para ter uma previsão de audiência para programas que já existem e que venham a ser lançados. O modelo preditivo que será entregue, deve receber alguns dados de entrada, sendo eles: data, horário e segmento do programa, com isso ele o software entregará uma previsão para o tipo de audiência, e tamanho da audiência medido em Rat%.

2.2. Justificativa

O grupo hefEStos propõe um modelo preditivo que entregará previsões se um programa específico terá ou não uma audiência adequada para tal(número de telespectadores), e também qual será o gênero, faixa etária e/ou a idade é o público alvo daquele programa piloto. Assim fazendo com que o cliente consiga otimizar seus gastos com programas, melhorar seu modelo preditivo e o modo como analisa seus dados .

3. Metodologia

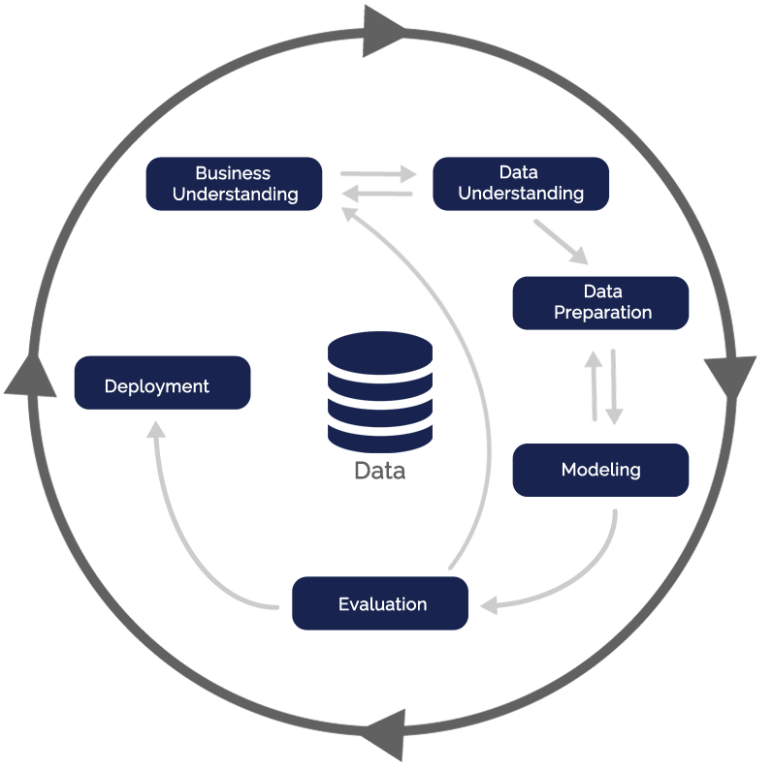
3.1. CRISP-DM

Descreva brevemente a metodologia CRISP-DM e suas etapas de processo

A metodologia CRISP-DM trata-se de um método de mineração de dados, na qual é descrita em termos de um modelo de processo hierárquico, consistindo em sets de tarefas, baseado em um conjunto de boas práticas para aplicação em Ciência de dados.

Dividida em alguns passos, a metodologia exige o entendimento do negócio, entendimento dos dados, a preparação e modelagem dos dados, além da avaliação do modelo e o deployment, conforme o Fluxograma 1. Tendo em vista que esses se distribuem nos 4 níveis de abstração: fases, tarefas genéricas, tarefas especializadas e instância de processo.

Fluxograma 1 - Metodologia CRISP-DM.



Fonte: MAGRATHEA (2018).

Tomando a primeira parte da metodologia, para o entendimento do negócio houve entrevista com o parceiro, além da apresentação de onboarding, na qual foi possível captar informações sobre a esfera em que o projeto se desenvolveria. Dessa forma, obteve-se o entendimento do funcionamento da análise de dados dentro da emissora corrente, somado a forma de como esses eram extraídos e utilizados, bem como, a identificação de prováveis vieses que poderiam existir nos dados.

A partir do primeiro contato, desenvolveu-se a etapa seguinte de entendimento dos dados. Após o recebimento desses, os mesmos foram analisados, avaliando o que significavam, onde entravam durante o processo e como seriam explorados. Assim, foi observado a necessidade de mais informações para a predição, haja vista que em primeira análise, só eram disponibilizados os valores de audiência total, audiência de cada perfil do público, alcance e fidelidade, mostrando um hiato no quesito de programação e categorias que os programas poderiam entrar.

Subsequente, na fase de preparação dos dados, em que esses são tratados, corrigidos e anulados em alguns casos de anomalias, de forma que seja mais benéfico para o projeto, é possível verificar diversas tarefas como: a mescla de datasets e registros; a seleção de subconjunto de amostra de dados; a agregação de registros; a derivação de novos atributos; a ordenação de dados para o modelo; a remoção ou substituição de valores nulos ou inválidos; o treinamento do modelo, etc. O pré-processamento dos dados do projeto em questão contou com a agregação de features e a seleção das que seriam realmente necessárias para avaliação, não necessitando de limpeza ou exclusão de registros, uma vez que os datasets disponibilizados estavam já corretos e limpos.

Na fase da modelagem, na qual o objetivo é determinar qual o modelo mais apropriado para o projeto, baseando-se no tipo de dado disponível para a mineração, o alvo da mineração e alguns dos requisitos específicos do modelo, houve o treinamento e teste de até 3 modelos diferentes, buscando pelo melhor índice de acurácia. Usando dos datasets tratados e anonimizados, esses foram expostos a cada um dos modelos, obtendo resultados diferenciados.

Para o diagnóstico desses resultados, a metodologia CRISP-DM conta com a fase de avaliação de modelo, que é quando se avalia a resposta obtida, usando de métricas de erro e outros artifícios, a fim de obter dimensões factuais da eficiência do modelo. Tratando do projeto do modelo preditivo para a audiência das emissoras, foram utilizadas especificamente métricas de avaliação de modelos de regressão, como o Erro Quadrático Médio (MSE), o qual penaliza mais erros maiores, já que os erros (diferença entre o valor previsto e o correto) são elevados ao quadrado, a Distância Absoluta Média (MAD), em que se faz a média do erro absoluto de cada previsão, além do R quadrado, que trata de uma métrica que varia entre $-\infty$ e 1 e é uma razão que indica o quão bom o modelo está em comparação com um modelo “naive”, que faz a predição com base no valor médio do target.

Seguidamente, das avaliações, se passa para o deployment, que trata-se da entrega e apresentação do produto para o stakeholder. É a partir dessa fase que a seguinte, feedback, é permitida. Na última fase, tem-se o retorno do cliente , na qual torna-se possível o aperfeiçoamento do produto final, dando caráter iterativo ao ciclo.

3.2. Ferramentas

- Colab: É uma plataforma em nuvem que simula um ambiente de programação com tudo pronto para o programador, no colab estamos colocando o core do projeto(o código do modelo preditivo)

- Python: É uma linguagem de programação de compreensão bastante acessível, com uma sintaxe simples e legibilidade clara, além de ter um vasto número de bibliotecas.

- Pandas: É uma biblioteca para uso em Python, open-source e de uso gratuito (sob uma licença BSD), que fornece ferramentas para análise e manipulação de dados.

- Matplotlib: É uma biblioteca para a visualização de dados em Python. Ele apresenta uma API orientada a objetos que permite a criação de gráficos em 2D de uma forma simples e com poucos comandos. A ferramenta disponibiliza diversos tipos de gráficos, como em barra, em linha, em pizza, histogramas entre outras opções.

- sklearn(scikit-learn): O scikit-learn é uma biblioteca da linguagem Python desenvolvida especificamente para aplicação prática de machine learning.Esta biblioteca dispõe de ferramentas simples e eficientes para análise preditiva de dados, é reutilizável em diferentes situações, possui código aberto, sendo acessível a todos e foi construída sobre os pacotes NumPy, SciPy e matplotilib.

- Numpy: é uma biblioteca para a linguagem Python com funções para se trabalhar com computação numérica

- Sheets: É um programa de planilhas incluído como parte do pacote gratuito de Editores de Documentos Google baseado na Web oferecido pelo Google.(É como se fosse uma planilha do excel online)

3.3. Principais técnicas empregadas

Descreva brevemente as principais técnicas empregadas, algoritmos e seus benefícios

- Label encoding
- One hot encoding
- Criação de dataframes com o uso da biblioteca pandas para a aplicação do modelo
- Métricas de erro e acurácia
- Uso da média dos valores
- Distribuição normal

Uma das técnicas empregadas foi a visualização por gráficos das informações, com o uso do matplotlib. Na criação de todos os gráficos, foi feita a seleção das features que estariam presentes no eixo X e eixo Y, então decidimos como queremos que ele nos retorna essa informação(gráfico de barras, frequência, distribuição normal...).

Outra técnica utilizada foram as ferramentas do sklearn(scikit-learn), das quais usamos para fazer algumas contas matemáticas, e principalmente para fazer e testar os modelos preditivos usados(KNN, LigthGMB e Regressão linear).

4. Desenvolvimento e

4.1. Compreensão do Problema

4.1.1. Contexto da indústria

Com mais de 500 funcionários, a Rede Gazeta é o maior grupo de comunicação do Espírito Santo, a empresa foi fundada em 1928, com o jornal A Gazeta, porém apenas no ano de 1976 a TV Gazeta surgiu, aproveitando o grande crescimento dos meios de comunicação em massa. Atualmente, eles contam com a presença de 16 veículos de comunicação, abrangendo a TV, rádio e internet.

As principais concorrentes deste mercado de comunicação no estado do Espírito Santo são a TV Tribuna, afiliada do SBT, e a TV Vitória, afiliada da Record. Na grande maioria do tempo a TV Gazeta se apresenta mais forte que suas concorrentes, salvo algumas exceções ocasionadas por eventos específicos.

4.1.2. 5 forças de Porter

Ameaça de produtos substitutos - apesar de ser quase impossível a substituição da TV, por já ser um meio consolidado há várias décadas, é observada a transferência de certa parcela do seu público para os meios digitais.

Ameaça de entrada de novos concorrentes - o único tipo de ameaça sofrido pela TV são os meios digitais, que contam com conteúdos “on demand”, ou seja, que o usuário consegue escolher o que vai assistir. Porém, agora comparando a RedeGazeta com as outras emissoras, sua vantagem se estabelece pela confiança dos telespectadores, sendo ressaltada através dos dados.

Poder de negociação dos clientes - seus clientes, como os patrocinadores, antigamente não possuíam um grande poder de barganha, por não haver nenhum canal de comunicação tão abrangente quanto a TV, porém, com a evolução dos meios digitais, essa competição ficou mais acirrada, com outros locais para veiculação de anúncios publicitários, aumentando o poder de barganha a esses anunciantes, e assim, forçando aos detentores do meio a baixarem o preço.

Poder de barganha dos fornecedores- os principais fornecedores de uma emissora, são as empresas que trabalham com equipamentos de áudio e vídeo, por exemplo. Já que existem poucas emissoras no estado, que tem uma relevância grande no mercado, as empresas terceirizadas possuem baixo poder de barganha, uma vez que a demanda de consumo desses produtos não é tão alta. Dessa forma, há uma maior dependência dos fornecedores para com a emissora, do que o contrário.

Rivalidade entre os concorrentes - essa força se refere a necessidade de antecipar tendências e estar sempre atualizado com as novidades do mercado. A solução desenvolvida pelo grupo hefEStos atuará mais fortemente nesse aspecto de ficar a par das novidades do mercado, ajudando a reação do público na previsão do score de audiência de um novo programa.

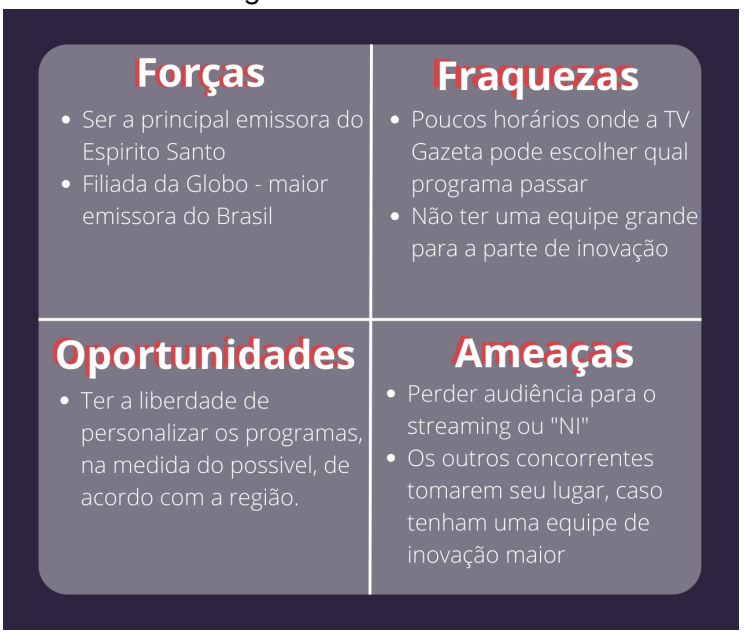
4.1.2. Análise SWOT

Com as análises feitas, foi possível verificar algumas forças, fraquezas, oportunidades e ameaças que o negócio apresenta. Dentro de *forças*, os principais pontos relacionados a essa parte da análise residem no fato de que a emissora parceira exerce um papel de líder no Espírito Santo, além de ser uma emissora afiliada da maior emissora do Brasil.

Já em *fraquezas*, fatores que se apresentam como ponto de melhoria frente ao negócio, pode-se inferir que os poucos horários nos quais a emissora parceira pode escolher qual programa exibir pode ser problemático, além disso, não ter uma equipe grande para a parte de inovação pode retardar o processo de renovação dos conteúdos programáticos para o público mais jovem e para o público ativo em mudança.

Considerando fatores externos, tratando de *oportunidades*, constata-se a possibilidade de conseguir personalizar os programas exibidos, na medida do possível, de acordo com a região, agradando mais a audiência e gerando mais engajamento. Ainda sobre elementos externos que podem afetar o negócio, tem-se as *ameaças*, em que se vê um cenário em que a empresa pode ter uma tendência a perder cada vez mais audiência para o Streaming ou “NI”, além disso, uma negligência com a equipe de inovação pode levar emissoras concorrentes, que investem muito nessa área, a se tornarem mais relevantes no mercado do que a emissora parceira.

Figura 1 - Matriz SWOT



Fonte: do próprio autor (2022).

4.1.3. Planejamento Geral da Solução

4.1.3.1. Dados disponíveis

Para o desenvolvimento da aplicação a Rede Gazeta disponibilizou alguns dados referentes à audiência, são esses:

- Datas;
- Hora de início;
- Emissoras;
- Dias da semana;
- Porcentagens utilizadas:
    - Rat%
    - Shr%
    - Rch%
    - Fid%
- Total de domicílios;
- Caracterização do público telespectador:
    - Classe:
        - AB;
        - C1;
        - C2;
        - DE;
    - Gênero:
        - Masculino;
        - Feminino;
    - Faixa etária:
        - 4-11 anos;
        - 12-17 anos;
        - 18-24 anos;
        - 25-34 anos;
        - 35-49 anos;
        - 50-59 anos;
        - 60+ anos.
- Grade de programação de cada emissora;

Divididos nas seguintes emissoras:

- Emissora 1;
- Emissora 2;
- Emissora 3;
- Canais pagos;
- Total Ligados Especial (TLE);
- Não identificado (NI);

Além disso, é dividido em dias da semana:

- Segunda a sexta;
- Sábado;
- Domingo.

4.1.3.2. Solução proposta

Sabendo da necessidade do parceiro por uma análise preditiva da audiência da Rede Gazeta, a solução proposta busca prover por meio de uma Inteligência Artificial, descrever a pontuação que determinado programa terá, quando lançado, de acordo com o horário, dia da semana, eixo e público especificado.

4.1.3.3. Tipo de tarefa

O tipo de método que será empregado será o de *Regressão,* pois iremos estimar dados de audiência de acordo com os valores de entrada que foram coletados anteriormente.

4.1.3.4. Como a solução deverá ser utilizada

O usuário deverá inserir os horários, a data, o segmento do novo programa e as características do público e a solução irá retornar uma previsão dos pontos de audiência, junto do peso de cada variável no modelo preditivo.

4.1.3.5. Quais são os benefícios trazidos

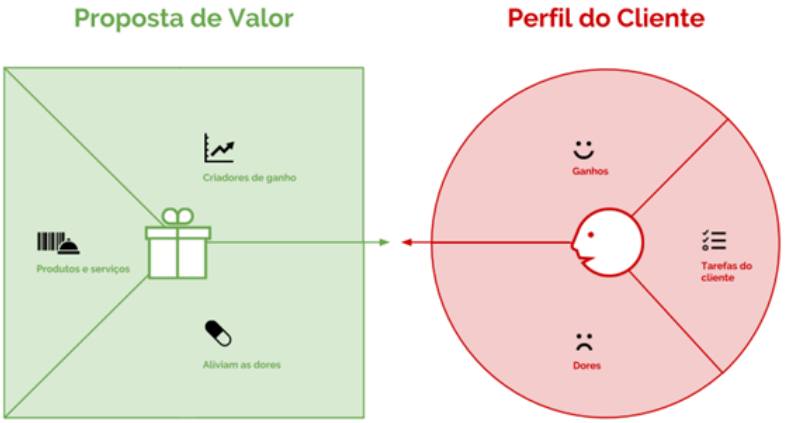
Os benefícios trazidos pela solução são os de melhorar a capacidade na realização da análise dos dados, de ajudar em previsões para lançamentos de programas futuros, além de promover uma melhora no desempenho de programas já existentes.

4.1.3.6. Qual será o critério de sucesso e qual será a medida utilizada

O critério utilizado como parâmetro de sucesso será a comparação com os valores fornecidos pelo banco de dados. Nesse, utilizamos parte do banco de dados para o treinamento do modelo, e o restante foi utilizado para comprovar o quão próximo foi o retorno do modelo com relação à realidade.

4.1.4. Value Proposition Canvas

Figura 2 - Value Proposition Canvas



Fonte: do próprio autor (2022).

4.1.4.1 Perfil do cliente

Tarefas do cliente:

- ➔ Criação de conteúdo para programas de TV;
- ➔ Analisar, de acordo com o histórico, se o programa iria se encaixar na programação.

Dores:

- ➔ Dificuldade em determinar o conteúdo adequado para agradar a audiência em um determinado horário;
- ➔ Alto investimento em programas que não repercutiram da forma esperada.

Ganhos do cliente:

- ➔ Melhora na acurácia da capacidade de prever a audiência de um determinado programa, baseado em algumas de suas informações como horário, data e tema. Adaptando, assim, o conteúdo, para aumentar o score de audiência.

4.1.4.2 Mapa de Valor

Produtos e serviços:

- ➔ Um modelo preditivo que recebe informações básicas de um possível novo programa,

e retorna um score de audiência e as principais variáveis que pesaram nesse.

Analgésicos/alívio das dores:

➔ Previsão acurada do possível sucesso de conteúdos, antes da produção.

Criadores de ganhos:

➔ Evita gastos com programas de baixa audiência;
➔ Fornece informações de conteúdos capazes de maximizar a audiência.

4.1.5. Matriz de Riscos

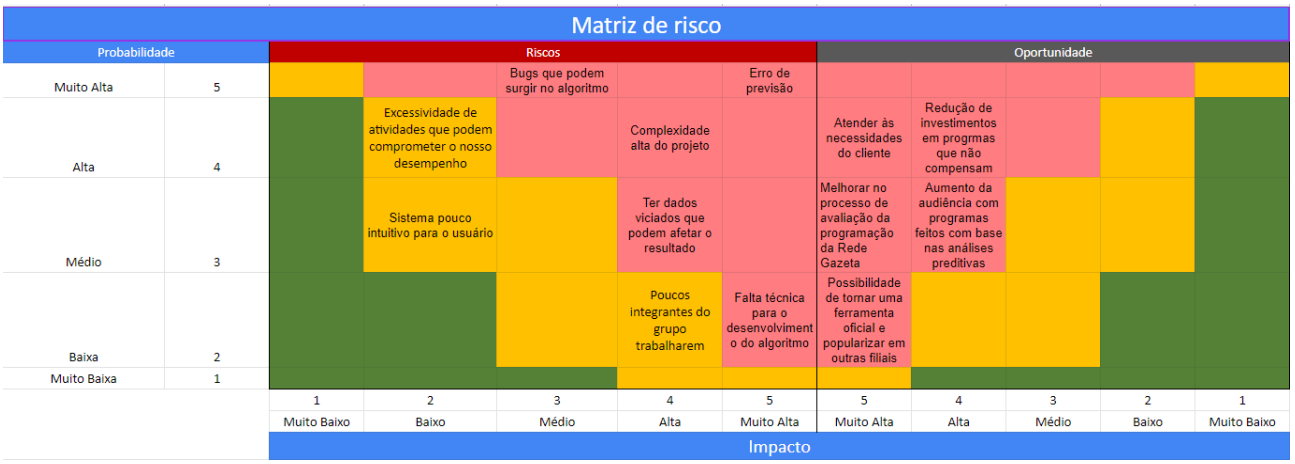
Para o desenvolvimento do projeto foi observado alguns riscos que poderiam ocorrer, sendo eles classificados entre negativos (riscos) e positivos (oportunidade). Cada um desses foram posicionados de acordo com seu impacto e probabilidade de acontecer, sendo os de vermelho os de maior preocupação e os verdes os menos preocupantes.

Reconheceu-se como riscos de alta probabilidade e alto impacto possíveis bugs que poderiam surgir no algoritmo, como consequência, erro de previsão, além do risco de complexidade alta do projeto, que pode afetar no sucesso do produto final.

Ademais, identificou-se outros riscos de menor gravidade, como excessividade de atividades que podem comprometer o desempenho da equipe, somado a possibilidade de poucos integrantes trabalharem, havendo uma concentração de tarefas, e a criação de um sistema pouco intuitivo para o usuário.

Como oportunidades, determinou-se com alta possibilidade e impacto a redução de investimentos em programas que não compensam para a emissora, além da grande chance de atender às necessidades do cliente, melhorando o processo de avaliação da programação, podendo sofrer aumento da audiência com programas feitos baseados nas análises preditivas e, a longo prazo, podendo se tornar uma ferramenta com escalas maiores que apenas a aplicação inicial do parceiro.

Figura 3 - Matriz de risco.

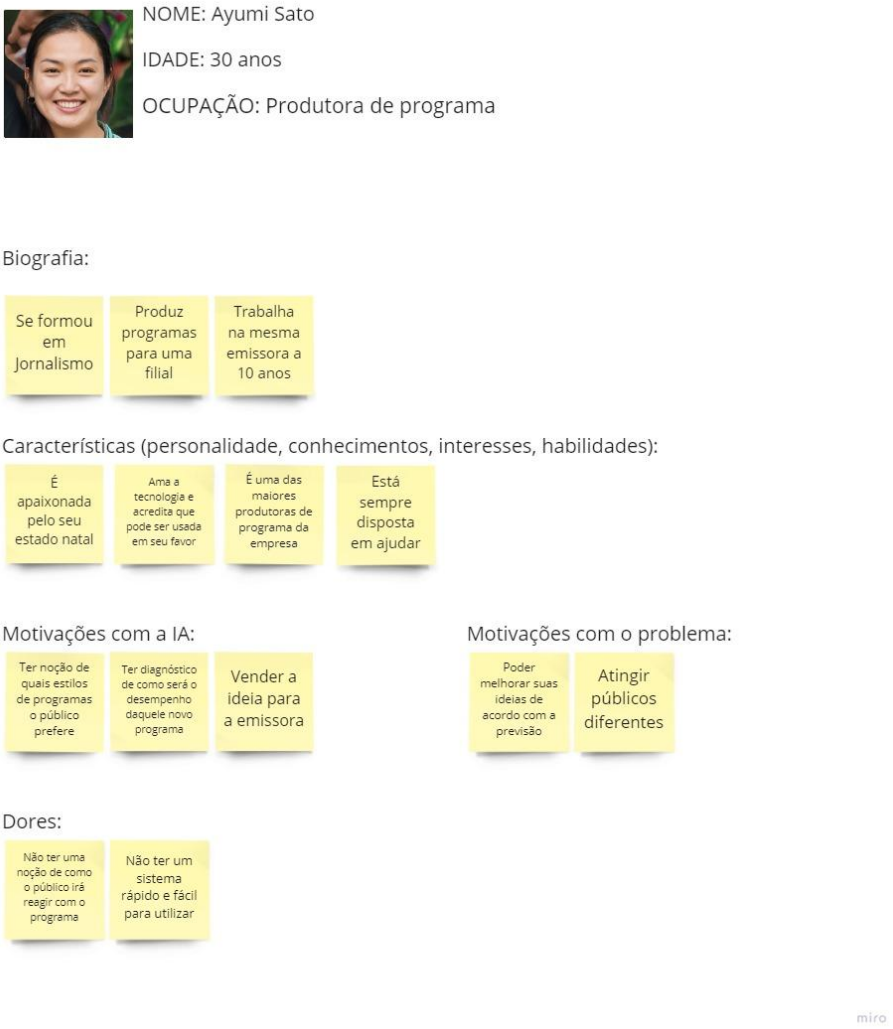


Fonte: do próprio autor (2022).

4.1.6. Personas

Pensando no público que se buscava atender, foram feitas duas personas, na qual uma representaria o público que usaria o produto para a produção de novos programas, pensando na audiência, público e horário. E a outra, tem como objetivo dinamizar a logística da análise de dados da emissora e aumentar o desempenho da grade de programação.

Figura 4 - Persona



Fonte: do próprio (2022).

Figura 5 - Persona

NOME: Arthur Silva

IDADE: 35

OCUPAÇÃO: Gerente de Operação

Biografia:

- Engenheiro de computação
- Trabalha em uma filial de uma emissora
- Trabalha na mesma emissora a 7 anos

Características (personalidade, conhecimentos, interesses, habilidades):

- Um apaixonado por tecnologia
- Sempre amou consumir TV aberta
- Não se adaptou com Streaming
- Apaixonado por matemática e dados

Motivações com a IA:

- Conseguir prever quais programas devem ter um maior investimento
- Análise de desempenho
- Mostrar para os patrocinadores a previsão de impacto desse programa

Motivações com o problema:

- Poder investir mais em programas que podem ter um futuro melhor
- Atingir públicos diferentes
- Maiores patrocínios

Dores:

- Não ter uma noção de como o público irá reagir com o programa
- Dificuldade de analisar os dados e receber algum diagnóstico
- Não ter uma noção qual público será atingido

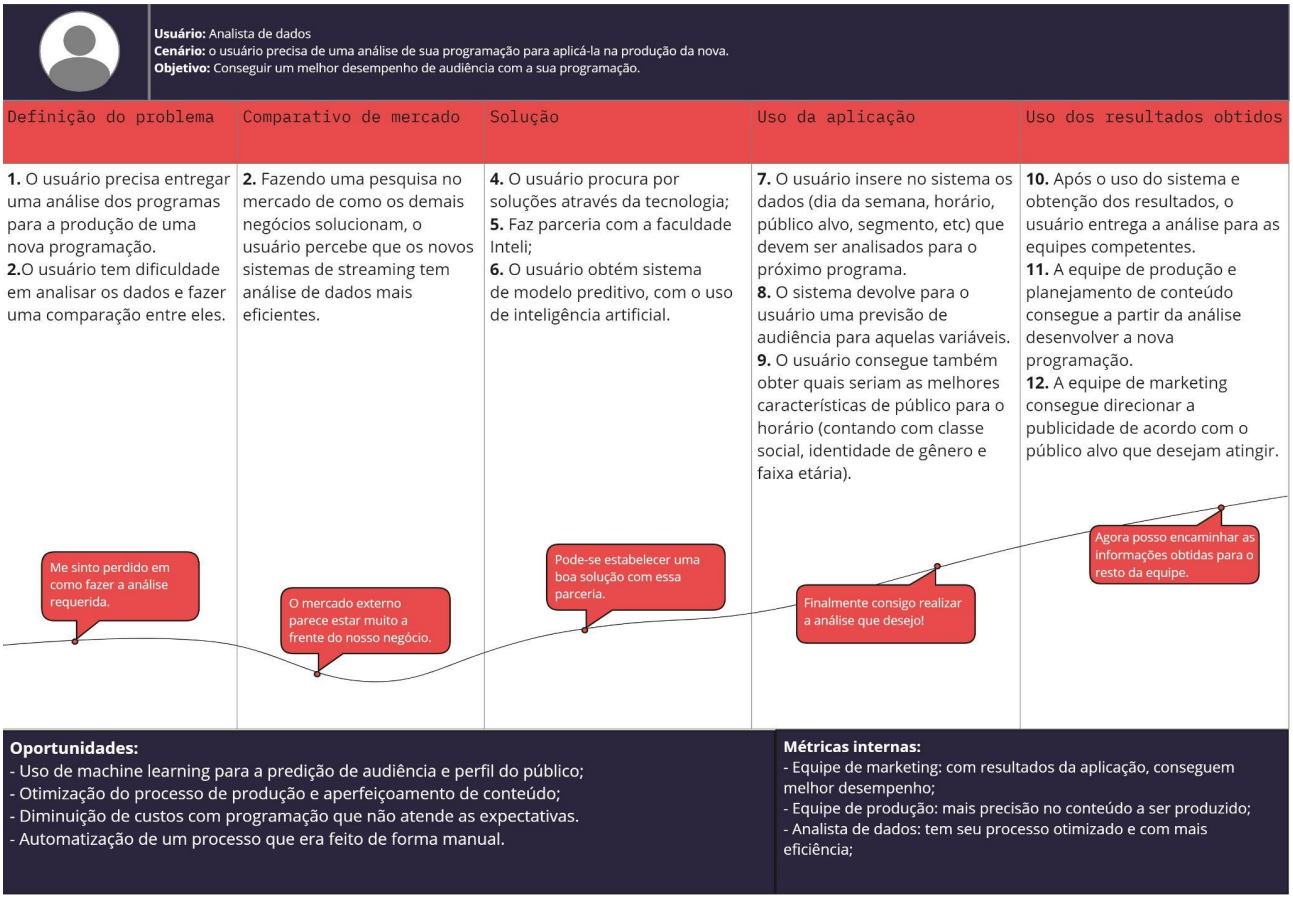
miro

Fonte: do próprio (2022).

4.1.7. Jornadas do Usuário

Sendo a jornada de usuário um mapa visual de todo o caminho traçado pelo cliente do antes, durante e depois do produto adquirido, tratando do corrente, foi considerado o período de definição do problema, quando o usuário identifica o contratempo a ser resolvido, seguido da fase de comparação de mercado, na qual o cliente busca por opções existentes dentro do mercado, passando para a fase de encontro da solução, quando o usuário encontra pelo produto produzido pela equipe *hefEStos*, e então segue para o uso da aplicação. Após o uso da AI produzida, os resultados obtidos são, finalmente, aplicados em análises e produção de novos programas.

Figura 6 - Jornada do usuário.



Fonte: do próprio (2022).

4.2. Compreensão dos Dados

4.2.1. Descrição dos dados

Recebemos dados de diferentes emissoras no formato XLSX, que foi convertido posteriormente para CSV, baseados em pesquisas do IBOPE, utilizando parâmetros percentuais em: Rat%, essa medida é calculada a partir da quantidade de indivíduos/domicílios ligados em determinado evento de TV, sendo que 1 ponto de audiência se equivale a 1% do universo pesquisado. Esse é calculado da seguinte forma: Rat% = (Rat# / universo) x 100; Shr% descreve a participação da audiência em um determinado evento, sobre o total de televisores ligados, em um determinado período. Esse é calculado da seguinte maneira: Shr% = (Rat% / TLE%) x 100 ; Rch% é o total de indivíduos, ou domicílios, diferentes alcançados por pelo menos 1 minuto. Reforçando que o tempo total, nesse caso, não está sendo considerado, e sim o contato que houve com a programação, faixa horária, emissora, etc. Esse é calculado da seguinte forma: Rch% = número de telespectadores / universo x 100; Fid% ilustra a permanência dos telespectadores no evento em questão, ou seja, quanto tempo daquele programa foi consumido pelo público. Esse é calculado da seguinte maneira: Fid% = Rat% / Rch% x 100. Além disso, recebemos informações acerca do perfil da audiência, como gênero, faixa de idade e classe social.

4.2.1.1. Descrição de como os dados serão agregados/mesclados

Devido ao alto volume de dados, para melhor visualização, esses foram mesclados usando a média dos valores, e agregados a partir de uma funcionalidade da plataforma google sheets. Ademais, na segunda semana do projeto, depois de múltiplos pedidos da turma, foi adicionado uma nova planilha no nosso conjunto de dados, que contempla a grade horária, com as seguintes informações: praça, data, faixa horária e o nome do programa e segmento das três emissoras que estamos trabalhando(Emissora 1, 2 e 3). Com essas novas informações, vamos poder ter uma noção de qual programa está fazendo o maior sucesso, comparando com as outras emissoras, e assim analisar o porquê isso acontece, se é por conta do horário ou pela falta de concorrência, etc.

4.2.1.2. Descrição dos riscos e contigências relacionados a esses dados

Nesse contexto, durante a análise dos dados, foi verificado que há dois riscos a serem considerados na análise de tais dados: o primeiro risco é os dados tenham viés, já o segundo se trata da concorrência injusta na coleta dos dados.

No primeiro ponto é necessário ponderar que, no aparelho utilizado na medição da audiência, quando colocado no domicílio de cada família, é criado um perfil para cada integrante, com informações de classe social, gênero e idade. Quando a TV é ligada, quem está assistindo deve selecionar o seu próprio perfil, é nesse momento que pode acontecer o viés, já que uma criança, por exemplo, pode selecionar errado ou mais de uma pessoa pode estar vendo

TV. Adicionalmente, a TV pode continuar ligada em um perfil originalmente correto, mas que já não é válido, ou seja, outro integrante pode ter começado a ver TV.prestando atenção e não foi selecionado, ou até mesmo uma mãe/pai pode, na pressa, selecionar o perfil próprio para uma criança assistir.

O segundo risco é por conta da concorrência injusta, já que a emissora chamada de Não Identificado, abrange algo muito genérico, isto é, apesar de ser chamada de uma emissora, dentro dela ela é dividida por diversas plataformas de vídeos, tornando a comparação injusta para as emissoras 1,2 e 3, que estão separadas.

4.2.1.3. Descrição de como será selecionado o subconjunto para análise inicial

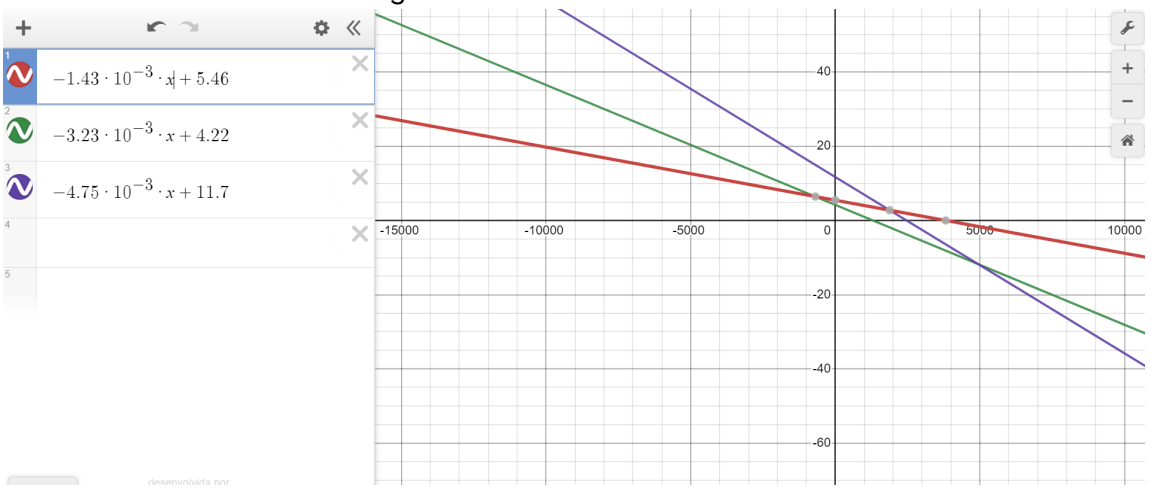
———————

4.2.1.4. Descrição das restrições de segurança

Para a elaboração do modelo, embora tenha sido concedido os dados de programação e audiência de algumas emissoras de TV aberta do Espírito Santo para a criação do modelo, foi proibida a publicação dos nomes das emissoras.

4.2.2. Descrição estatística básica dos dados

Figura 7 - Gráfico de tendência



Fonte: do próprio (2022).

- ➔ Emissora 1 = azul
- ➔ Emissora 2 = verde
- ➔ Emissora 3 = vermelho

Este gráfico ilustra a tendência de audiência nas emissoras em um intervalo de 2 anos, pode ser observado uma queda forte de audiência em todas, se destacando a maior queda na emissora 1.

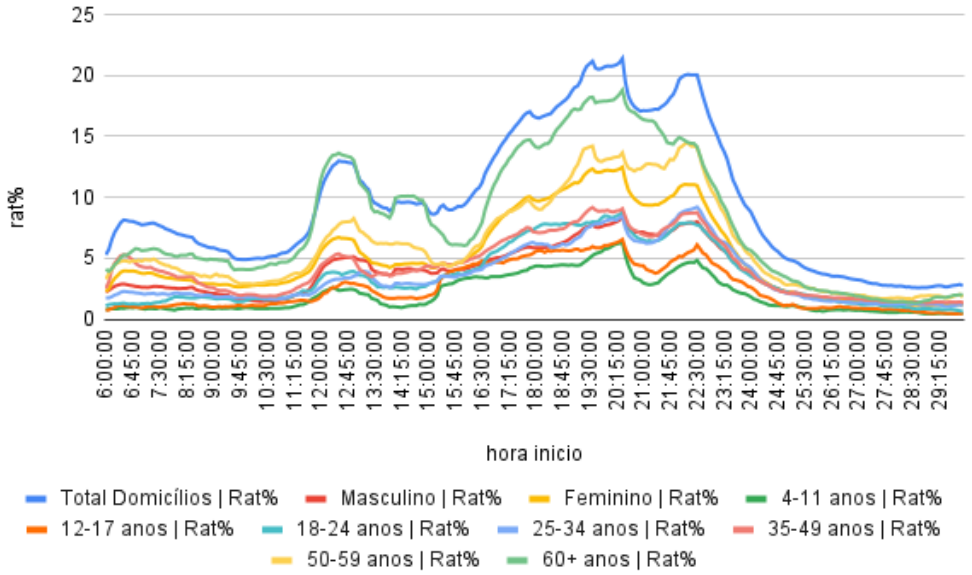
4.2.3. Dicionário dos dados

Parâmetros	Descrição	Como calcular
Rat%	Quantidade de indivíduos/domicílios ligados na TV (1 ponto = 1%)	Rat% = (Rat# / universo) x 100
Shr%	Participação da audiência em um evento, sobre o TLE de um período	Shr% = (Rat% / TLE%) x 100
Rch%	Total de domicílios ou indivíduos alcançados por 1 minuto ou mais	Rch% = número de telespectadores / universo x 100
Fid%	Permanência dos telespectadores naquele evento	Fid% = Rat% / Rch% x 100

4.2.4. Emissora 1

Segunda a sexta

Figura 8 - Audiência de segunda à sexta da emissora 1.



Fonte: do próprio autor (2022).

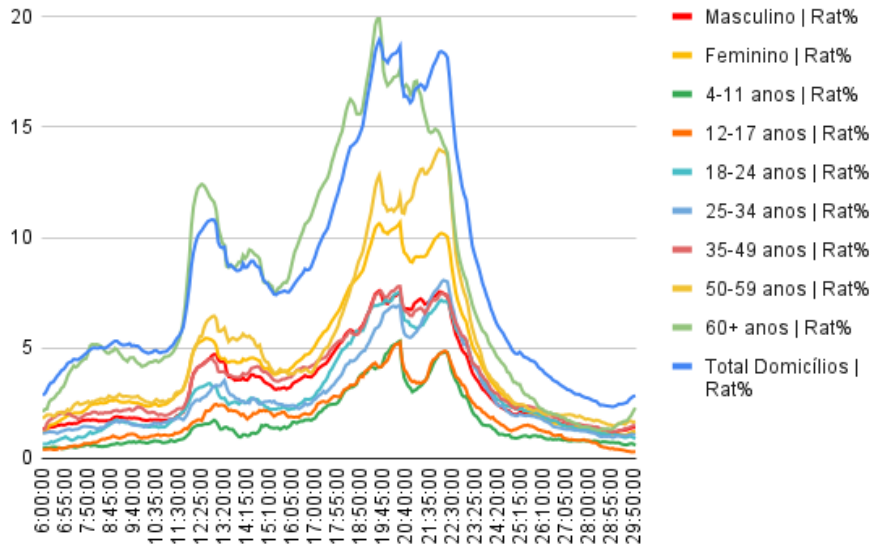
O primeiro gráfico que fizemos é da Emissora 1, juntando todos os dados de segunda a sexta, entre junho de 2020 e junho de 2022. Dividimos o Rat% em segmentos de idade, gênero e total de domicílios, e todos os dados são coletados a cada 5 minutos.

Analisando o gráfico, percebemos que em todos os horários o público feminino é predominante em comparação com o masculino. Seguindo a mesma ideia, o público com mais de 60 anos é predominante em comparação ao resto das idades, se mantendo estável durante toda a tarde, o único momento em que eles perdem a soberania é depois das 22:00, onde o público de 50-59 anos ganha.

Existem dois picos claros nesse gráfico: 1. No horário do almoço(aproximadamente às 12h), no momento em que está passando programa jornalístico regional. 2. Durante a noite (das 18h até aproximadamente 00h), que é o horário em que são exibidos programas de entretenimento como novelas e reality shows e jornais. Pode-se notar também que no horário de exibição de um jornal à noite (aproximadamente das 20h30 às 21h30), há uma redução do público.

Sábado

Figura 9 - Audiência de sábado da emissora 1.



Fonte: do próprio autor (2022)

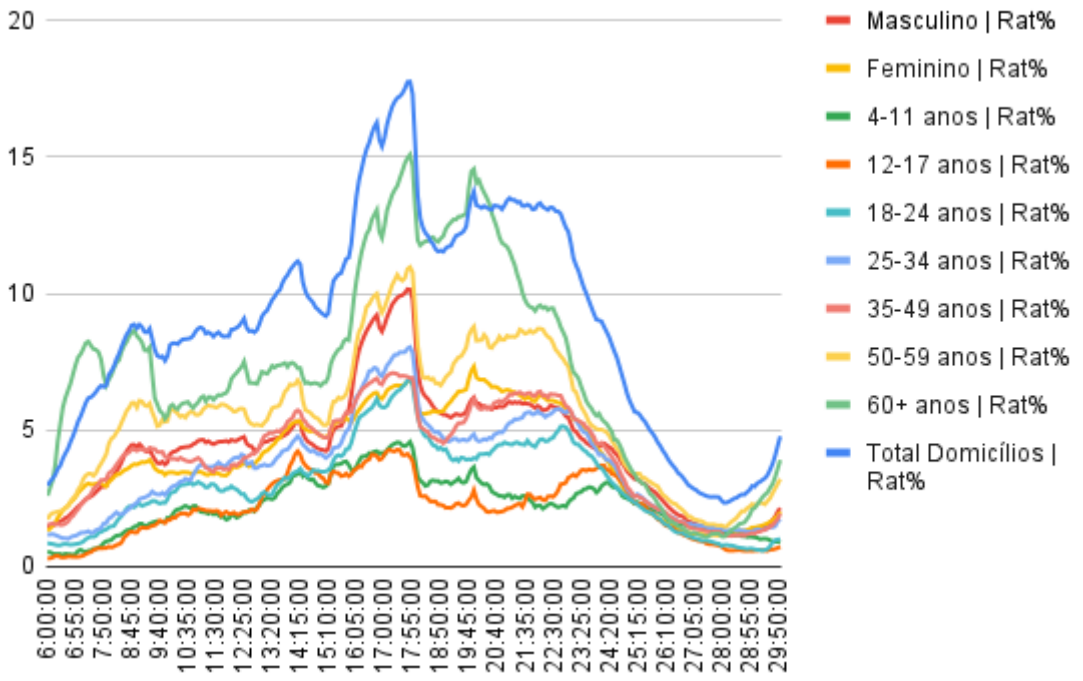
Fizemos um gráfico que reúne todos os sábados do mesmo espaço de tempo (junho de 2020 até junho de 2022), dividimos nos mesmo segmentos do gráfico anterior: todos os dados de Rat% com gêneros e idades e o total de domicílios.

Seguindo o mesmo padrão do gráfico anterior, as pessoas que têm mais de 60 anos continuam na soberania de audiência, só que percebemos que no sábado a diferença é bem menor comparado aos dias úteis. Além disso, em alguns horários essa faixa etária ganha do total de domicílios, como por exemplo entre 14:15 até 18:50.

As mulheres continuam com uma maior audiência do que homens, seguindo o mesmo padrão do gráfico anterior, porém em alguns horários, como 14:00 e na madrugada, essa diferença diminui bastante.

Domingo

Figura 10 - Audiência de domingo da emissora 1.



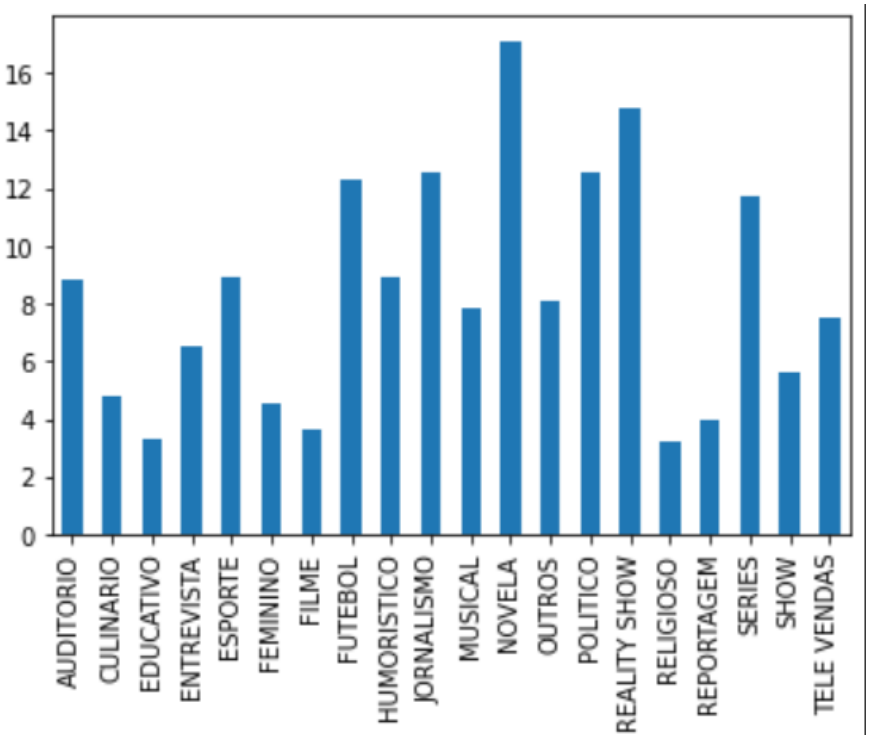
Fonte: do próprio autor (2022).

Fizemos um gráfico que reúne todos os domingos do mesmo espaço de tempo (junho de 2020 até junho de 2022), dividimos nos mesmo segmentos do gráfico anterior: todos os dados de Rat% com gêneros e idades e o total de domicílios.

Nesse gráfico, pode-se notar que a variação da audiência ocorre de outra forma, devido à diferença muito grande da grade de programação com relação aos dias úteis, sendo muito constante ao longo do dia com programações variadas de reality show, carros, rural,etc, exceto no horário do programa de futebol(das 15:50 às 18:05), em que há um pico em praticamente todos os nichos. Ainda nesse contexto, deve-se ponderar que logo após o término da programação de futebol há uma queda que leva a audiência de volta para o patamar anterior, e fica estável durante a noite com uma programação de auditório e de show.

Divisão por segmento

Figura 11 - Gráfico de audiência por segmento da emissora 1.



Fonte: do próprio autor (2022).

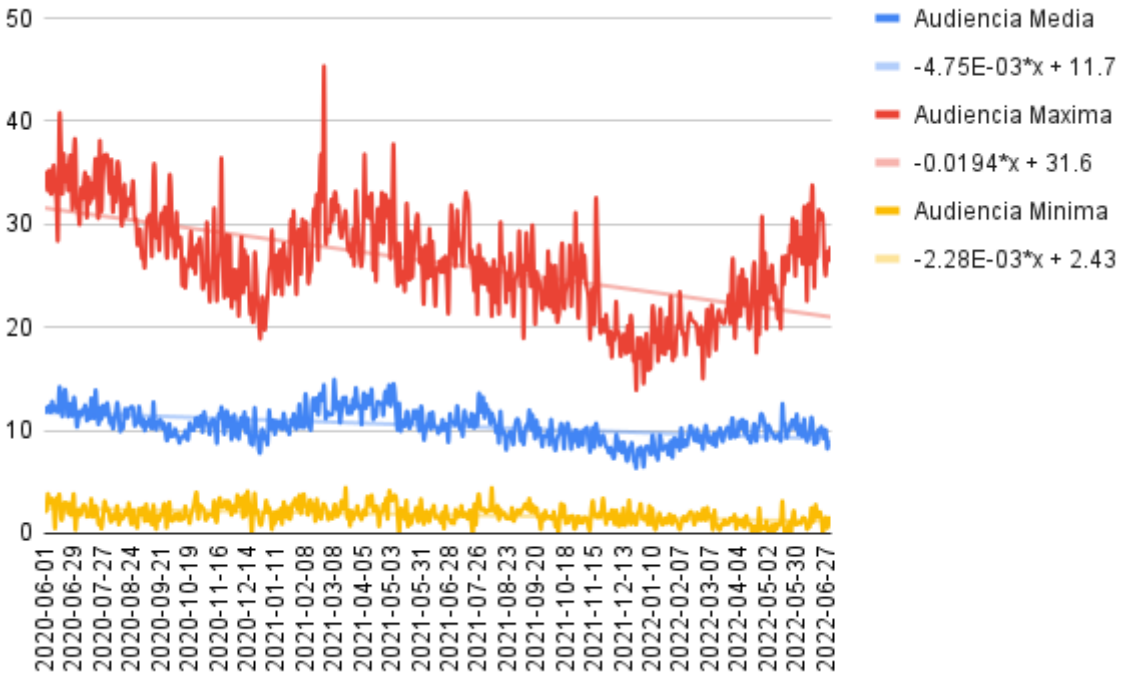
Quando recebemos os dados, cada programa estava separado em segmentos diversos, com isso fizemos um gráfico com o Rat% com o total de domicílios, assim os número no eixo y representam as porcentagens e o eixo x os segmentos. Esse gráfico representa somente os segmentos da emissora 1.

Podemos observar que os três segmentos com maior audiência são: novela, reality show, jornalismo e políticos, que estão empatados, e futebol, em ordem decrescente. Já os com pior audiência são: educativo, religioso, filme e reportagem, em ordem crescente.

Porém interpretando os dados podemos percebemos que essa alta de audiência no reality show pode acontecer por conta de um programa específico, que ocorre em alguns meses do ano, podem dar essa maximizada nos dados, sendo um ruído.

Segunda a Domingo - Audiência

Figura 12 - Gráfico de médias da audiência diária.



Fonte: do próprio autor (2022).

Esse gráfico nos mostra uma média de 3 dados importantes: audiência máxima, média e mínima. A primeira mede qual foi a maior quantidade de pessoas que estavam com a TV ligada na emissora 1 naquele dia específico, ou seja, o pico do dia. Já a segunda é a média de audiência naquele dia, que se mantém bem estável, comparando com o primeiro dado citado. O terceiro dado é exatamente o oposto do primeiro, então ele mede quanto que foi a menor quantidade de pessoas ligadas na TV naquele dia, ou seja, o ponto baixo do dia.

Podemos notar com clareza que, a audiência média e mínima, se mantém estável, quando comparado com a linha, que mostra a média daqueles valores. Já a audiência máxima isto não acontece, existem pontos específicos em que a audiência é visivelmente mais alta ou mais baixa. Notamos que esses picos mais altos são em momentos particulares, como final de algum reality show, final de campeonato de futebol, ou em feriados como o Natal.

4.2.4. Emissora 2

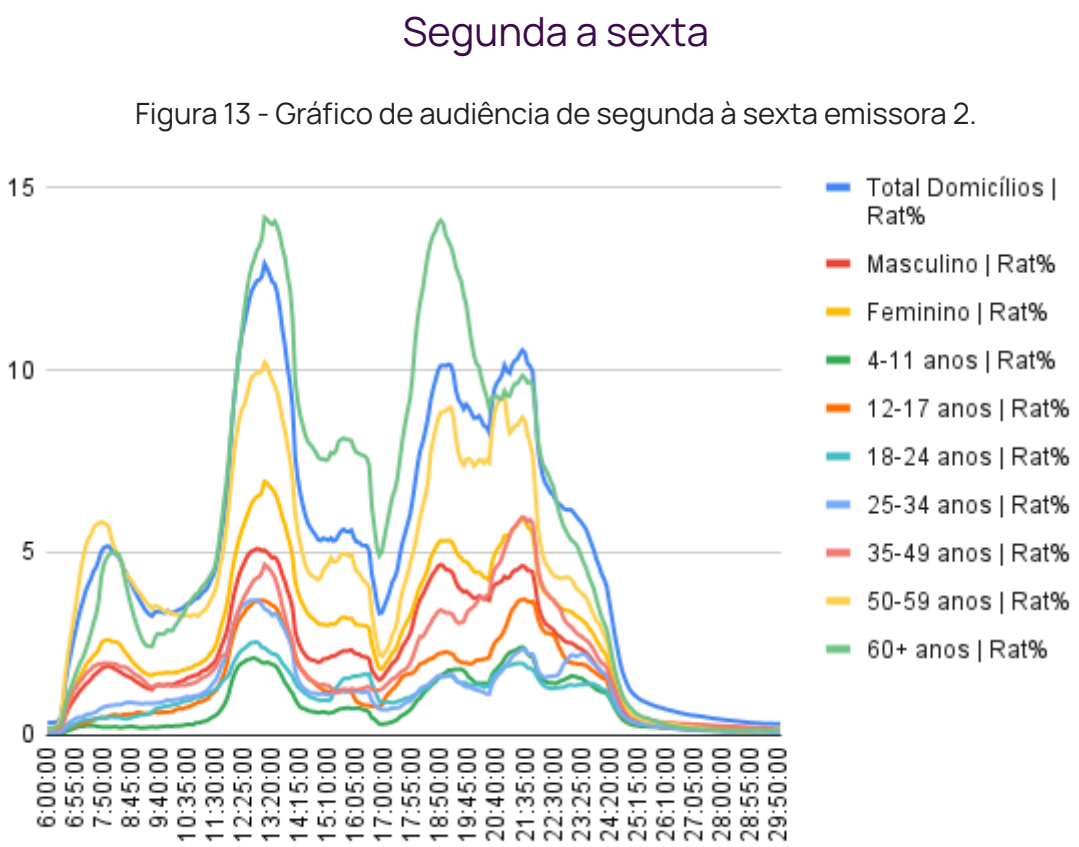


Figura 13 - Gráfico de audiência de segunda à sexta emissora 2.

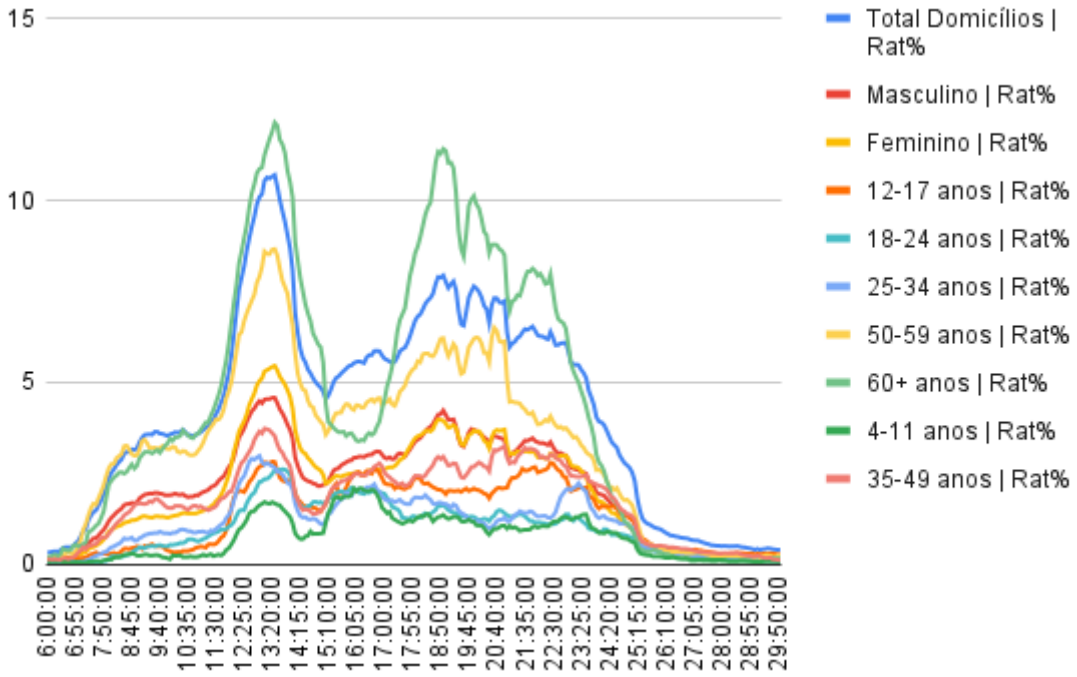
Fonte: do próprio autor (2022).

O gráfico acima representa a média de audiência média dos dias de segunda a sexta dos últimos dois anos na emissora 2. Sendo separado por idade, gênero e total de domicílios. Como a maioria dos gráficos que analisamos, esse se mantém no mesmo padrão: o sexo feminino ganha do sexo masculino em todos os horários, e o mesmo acontece com a idade 60+, porém algo se diferencia: as pessoas que tem 50-59 anos ganham de todas as outras idades, menos de 60+, e os dois gêneros e em alguns momentos até mesmo do total.

Os picos principais acontecem durante o horário de almoço (12:25 - 14:15) e no início da noite (18:50 - 20:40) e isso se dá em todas as linhas, em proporções diferentes. Por outro lado existem momentos que há uma queda muito grande, 14:15 - 15:10 e 17:00, isso pode ser por dois motivos: um deles é que as concorrentes podem estar passando algo que é mais interessante para esse público ou o programa que passa na emissora 2, nesses horários, não agrada o público.

Sábado

Figura 14 - Gráfico da audiência de sábado da emissora 2.

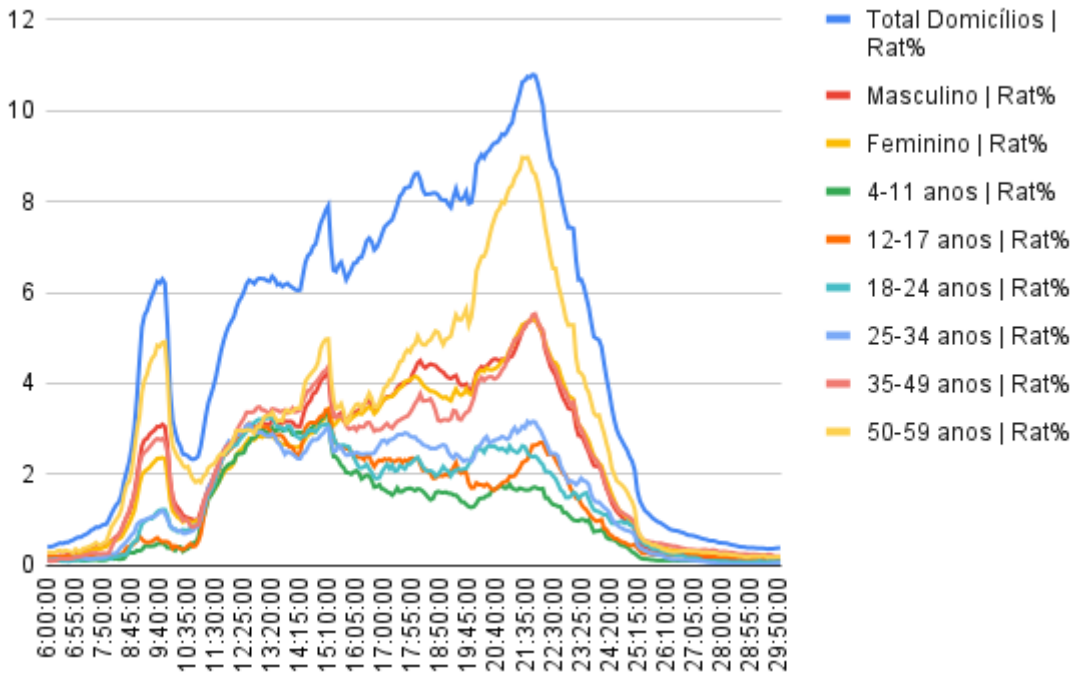


Fonte: do próprio autor (2022).

Este gráfico mostra a média de audiência por horário nos sábados da emissora 2 dividido por faixas etárias. Pode-se perceber que ao longo do dia inteiro a audiência predominante é de 50+ anos. Nota-se também dois picos de audiência, o primeiro às 13:20h e o segundo por volta das 19:00h, sendo o segundo mais dispersado. Os picos do gráfico são muito mais definidos nas faixas etárias mais velhas, ou seja, apesar de o público jovem ter menor audiência em todos os horários sua audiência é mais constante ao longo do dia.

Domingo

Figura 15 - Gráfico de audiência de domingo da emissora 2.



Fonte: do próprio autor (2022).

Juntando todos os dados de Domingo, entre junho de 2020 e junho de 2022. Dividimos o Rat% em segmentos de idade, gênero e total de domicílios, e todos os dados são coletados a cada 5 minutos.

Analisando o gráfico, percebemos que em todos os horários o público feminino é predominante em comparação com o masculino. Seguindo a mesma ideia, o público com mais de 60 anos é predominante em comparação ao resto das idades, se mantendo estável a partir do 12:00 durante toda a tarde e à noite, o único momento em que eles perdem a soberania é depois das 7:00, onde o público de 50-59 anos ganha.

A audiência é mantida alta das 12:00 até 23:00 em média, possuindo um pico às 13:00, que pode ser em decorrência da hora média do almoço. Sendo o único momento de baixa audiência durante a madrugada e de manhã até às 12:00.

4.2.5. Conteúdo Não Identificado

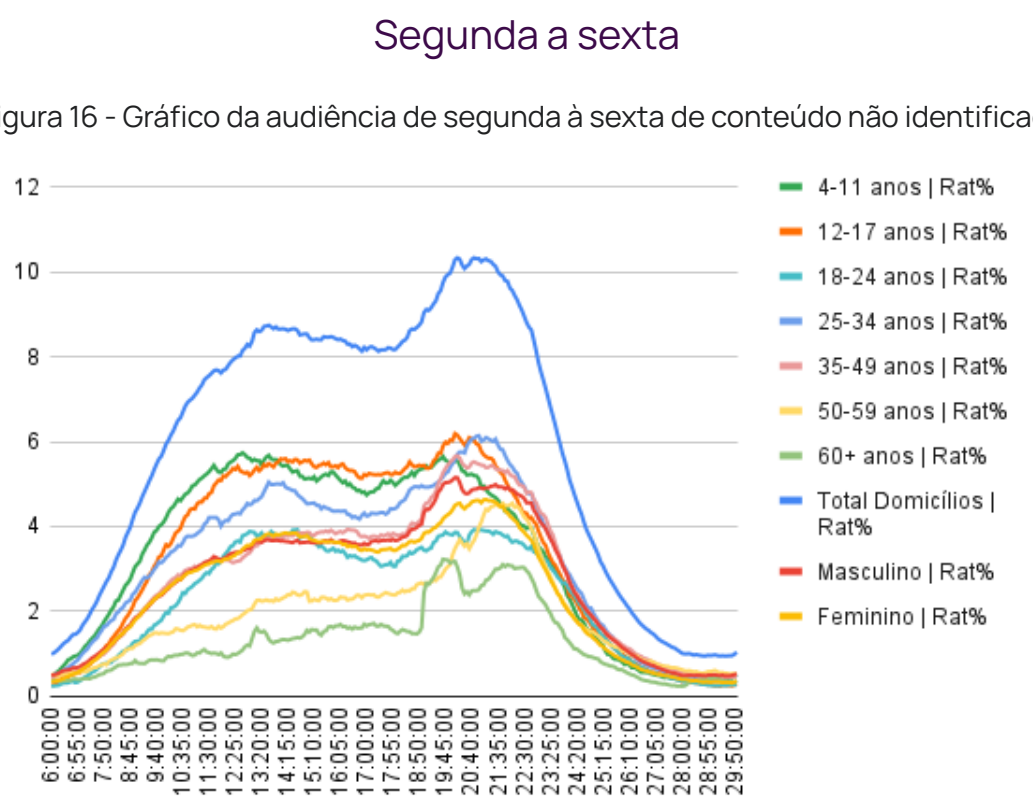


Figura 16 - Gráfico da audiência de segunda à sexta de conteúdo não identificado.

Fonte: do próprio autor (2022).

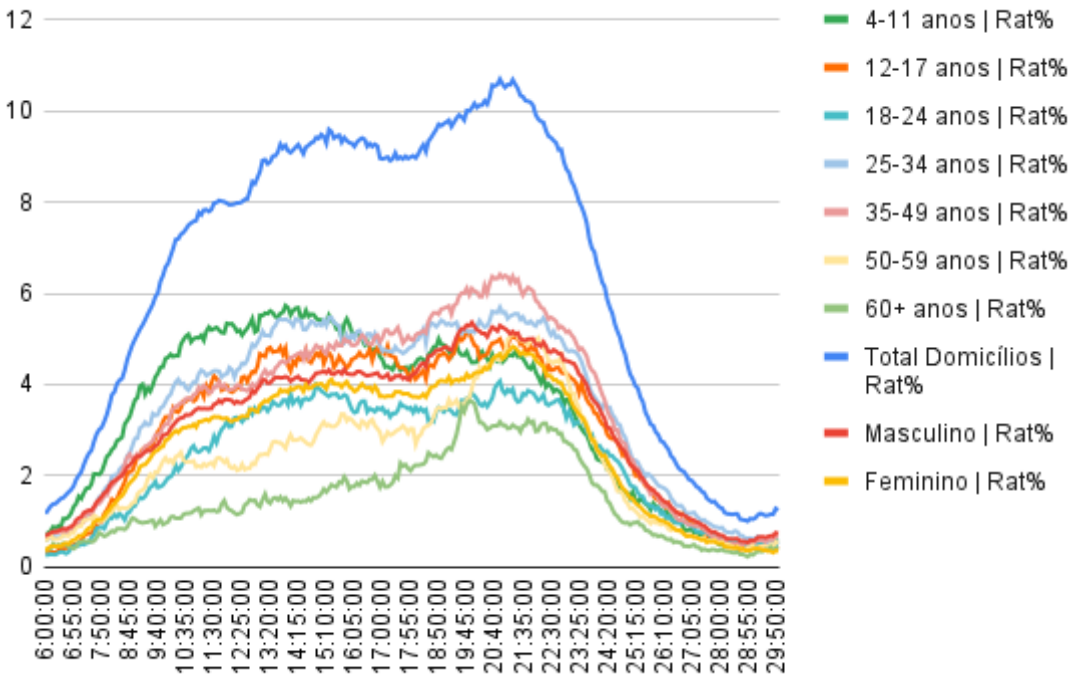
O primeiro gráfico que fizemos é a junção de todos os dados de segunda a sexta, entre junho de 2020 e junho de 2022. Dividimos o Rat% em segmentos de idade, gênero e total de domicílios, e todos os dados foram coletados com um intervalo de 5 minutos.

Analisando o gráfico, é possível perceber que a partir das 19:00 até 00:00 o público masculino é predominante, outro ponto que pode se observar é que o público entre 12-17 anos é predominante em todos os horários.

Existem dois picos claros nesse gráfico: 1. No final da tarde (por volta das 18:00) o único público que tem um pico menos relevante são as pessoas de 18-24 anos. 2. Durante a noite (por volta das 20:40)nos públicos com 50-59 anos e 60+ anos, também se vê um pico não muito relevante entre as pessoas com 18-24 anos.

Sábado

Figura 17 - Gráfico de audiência de sábado do conteúdo não identificado.



Fonte: do próprio autor (2022).

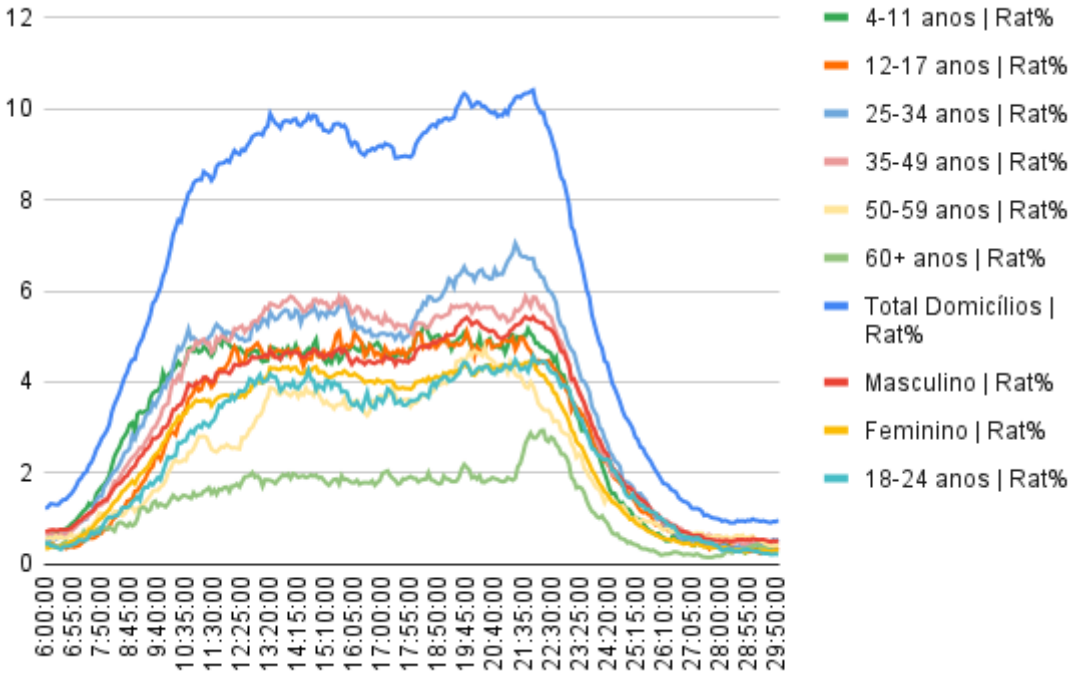
Fizemos um gráfico que reúne todos os sábados do mesmo espaço de tempo (junho de 2020 até junho de 2022), dividimos nos mesmo segmentos do gráfico anterior: todos os dados de Rat% com gêneros, idades e o total de domicílios.

Diferente do gráfico anterior, o público masculino no se mantém predominante durante o dia inteiro aos sábados, outro ponto é que o público de 4-11 anos se mantém predominante durante o período da manhã até a tarde(por volta de 16:40) que é quando o público de 35-39 anos vira o público predominante durante o resto do dia.

Também é possível perceber um aumento de audiência na maior parte dos públicos por volta de 19:45, menos no público infantil(4-11 anos) que tem um leve pico de audiência que depois começa a cair .

Domingo

Figura 18 - Gráfico de audiência de domingo do conteúdo não identificado.



Fonte: do próprio autor (2022).

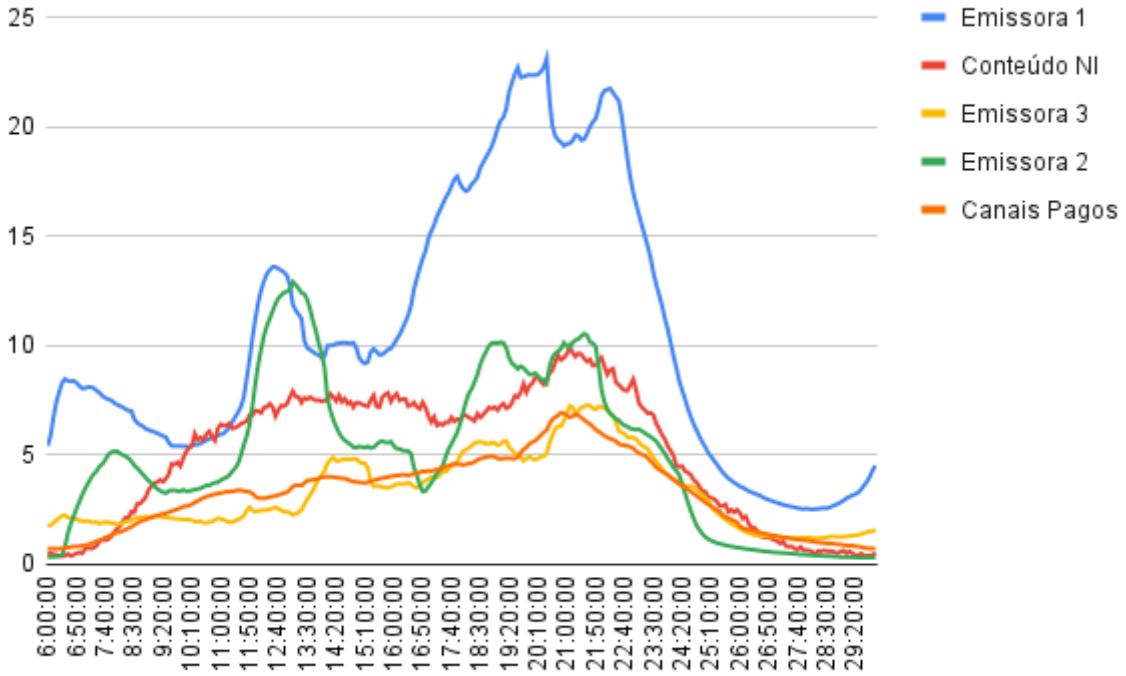
Fizemos um gráfico que reúne todos os domingos do mesmo espaço de tempo (junho de 2020 até junho de 2022), dividimos nos mesmo segmentos do gráfico anterior: todos os dados de Rat% com gêneros, idades e o total de domicílios.

No gráfico de domingo, podemos perceber que a audiência do público de 60+ anos é a menor de todas e se mantém assim durante o dia inteiro, tendo um leve pico no período das 21:45, outro ponto interessante é que o público de 50-59 anos começa o dia com um audiencia bem diferente das outras faixas etárias e por volta de 16:50 começa a se assemelhar com os outros públicos.

Já os públicos de outras faixas etárias se mantêm semelhantes durante o dia inteiro, tendo um aumento de audiência no final da tarde(por volta de 17:40), e começa a ter uma queda de audiência por volta das 22:40. Algo que pode ser percebido nesse intervalo de tempo é que o aumento de audiência no público de 24-35 anos é o mais relevante entre esses públicos.

4.2.6. Comparação entre emissoras

Figura 19 - Gráfico de comparação de audiência entre as emissoras.



Fonte: do próprio autor (2022).

É possível observar uma predominância de audiência da emissora comparada a todas as outras, sendo que a emissora 4 durante o período de almoço tem uma audiência equilibrada com ela, mas durante o resto do dia é menor.

Nesse gráfico também notamos que a emissora 1 e 4 tem um pico de audiência por volta de 12:40, a emissora 1 também tem picos de audiência durante o período da noite mas que se difere, das demais emissoras(por volta de 20:35 e 22:40), a emissora 4 também tem picos durante o período da noite(por volta de 18:55 e 21:25). As demais emissoras têm um constante crescimento de audiência ao longo do dia até as 21:00 e depois disso sua audiência começa a cair e só volta a se recuperar no período da manhã por volta das 10:10.

O aspecto mais marcante desse gráfico é que a emissora 1 não perde quase nunca para nenhuma outra emissora, somente entre 10:10 - 11:50 (contéudo NI) e 13:20 - 14:20 (emissora 2), fora esse ponto, quase não há concorrência para a emissora que foi primeira citada.

4.2.7. Descrição da predição desejada, identificação da sua natureza

A natureza da predição desejada é contínua, uma vez que a saída dos dados será feita através da predição de um score de audiência.

4.3. Preparação dos Dados

4.3.1. Descrição das manipulações necessárias nos registros e suas respectivas features

4.3.1.1. Tabelas das Emissora 1 e 2 - Grade Horária (Seg a Sex, Sab e Dom) com Audiência (Seg a Sex, Sab e Dom)

Etapa de Anonimização:

CSV Grade Diária (Seg a Sex, Sáb, Dom)

Para a anonimização dessas tabelas, alteramos os nomes das emissoras nas colunas 5, 6 e 7 para Emissora 1, Emissora 2, Emissora 3.

Código 1- Anonimização dos programas

```
#Ajustes no dataset da emissora 1 de segunda a sexta e sábado e domingo
dataSet = dataSet.drop(['Programa'], axis=1)
```

No Código 1, excluímos a coluna “Programas” das tabelas de cada emissora. Para isso, usamos o comando “.drop”.

Código 2 -Anonimização das categorias e criação da categoria “BBB”

```
#dicionario para podermos manipular os dados da coluna categoria no pandas
dict_categoria = {'JORNALISMO' : '14', 'AUDITORIO' : '1', 'FILME' : '10', 'NOVELA' : '18', 'SERIES' : '27',
       'ENTREVISTA' : '7', 'REALITY SHOW' : '22', 'HUMORISTICO' : '13', 'REPORTAGEM' : '25',
       'EDUCATIVO' : '6', 'DOCUMENTARIO' : '5', 'FUTEBOL' : '11', 'ESPORTE' : '8', 'POLITICO' : '20',
       'SHOW' : '28', 'FEMININO' : '9', 'DEBATE' : '4', 'MUSICAL' : '16', 'RELIGIOSO' : '24', 'SORTEIO' :
'29',
       'MINISSERIE' : '15', 'NAO CONSTA' : '17', 'OUTROS' : '19', 'CULINARIO' : '3', 'TELE VENDAS' : '30',
       'RURAL' : '26', 'CARROS E MOTORES' : '2', 'PREMIACAO' : '21', 'GAME SHOW' : '12'}
#Muda todos os nomes das categorias por numeros, pre definidos no dicionario acima
clean_merged.Categoria = clean_merged.Categoria.replace(dict_categoria)
```

```
                    #faz com que o programa BBB se torne um numero para podermos mudar sua
categoria
dict_realities = { 'BIG BROTHER BRASIL' : 23}
#muda na tabela o valor do programa BBB
clean_merged.Programa = clean_merged.Programa.replace(dict_realities)
#faz com que todas linha que o programa tem o numero 23(BBB) mude sua categoria para 23(BBB)
clean_merged.loc[clean_merged['Programa'] == 23, 'Categoria'] = 23
```

No código 2, trocamos os nomes das categorias por números de 1 a 30, e também fizemos com que toda vez que algum programa da categoria reality show estivesse passando "BBB" criássemos uma categoria diferente para ela(numero 23).

Etapa de Junção:

Código 2 - Grade Horária com Audiência (Emissora 1,2).

```
#unindo as duas tabelas das quais uma corresponde a informacoes de quem assiste e outra com
estilo do programa; com base no horario e dia em que passa
new_merged_emissora = pd.merge(rat_seg_sex1, data_seg_sex,how='left',left_on=['Data','Hora
Início'], right_on=['Data','Faixa Horária'])
```

Fonte: do próprio autor (2022).

Conforme o código 2 e 3, juntamos a tabela de Grade Horária com a de audiência para ter acesso à audiência que aquela categoria representa em determinado horário. Assim, podendo relacionar os dados a uma determinada categoria de programa no modelo preditivo.

Código 3 - Junção Seg a Sex com Sab e Dom (Emissora 1,2 e NI).

```
#Junta a tabela da semana(seg a sex), do sábado e de domingo
clean_merged3 = pd.concat([clean_seg_sex_streaming,clean_sabado_streaming,
clean_domingo_streaming])
#faz com que todos os dados da coluna data sejam convertidos para datetime um tipo de variavel
proprio do pandas
clean_merged3['Data'] = clean_merged3['Data'].apply(pd.to_datetime)
clean_merged3.drop_duplicates(inplace=True)
clean_merged3.sort_values(by=['Data','Hora Início'], ascending=True)
```

Fonte: do próprio autor (2022).

Após a junção da grade horária com a audiência de cada tabela (Seg a Sex, Sab e Dom) e seu tratamento (Agregação e Filtragem) decidimos juntar ela em uma tabela só. Além de juntá-las, organizamos elas em ordem de dia da semana e horário (.sort).

Etapa agregação:

Código 4 - Programação/Categoria (Emissora 1 e 2).

```
#separa uma informacao em duas colunas para melhor leitura da tabela
new_merged_emissora[['Programa','Categoria']] = new_merged_emissora['Emissora 1'].str.split(' / ', expand=True)
```

Fonte: do próprio autor (2022).

Para conseguirmos usar separadamente as informações que estavam dentro dessa coluna, dividimos ela em duas. Portanto, usamos o comando ".split" para dividir a coluna em duas novas colunas (Programa e categoria), pois na coluna da emissora a informação estava junta, tendo apenas uma "/" separando-as, fizemos isso para conseguirmos manipular melhor os dados

Código 5 - Data (Emissora 1 e 2).

```
#Separa a coluna data em 3 colunas "Ano, mes e dia"
clean_merged['Dia'] = pd.to_datetime(clean_merged['Data']).dt.strftime('%d')
clean_merged['Mes'] = pd.to_datetime(clean_merged['Data']).dt.strftime('%m')
clean_merged['Ano'] = pd.to_datetime(clean_merged['Data']).dt.strftime('%Y')
clean_merged
```

Fonte: do próprio autor (2022).

Para conseguir utilizar as informações presente na coluna "Data" tivemos que dividi la em três novas colunas "Ano", "Mês" e "Dia"Assim, usamos o ".strftime" para dividir "Ano-Mês-Dia" pelo valor correspondente a cada um deles e criar em três novas colunas para agrupar essas informações separadamente.

Etapa de filtragem :

Código 6 - Data = Hora início (Emissora 1 e 2).

```
#igualando duas colunas para unir-las
data_seg_sex['Faixa Horária'] = rat_seg_sex1['Hora Início']
```

Fonte: do próprio autor (2022).

Igualamos a coluna "Faixa Horária" com a "Hora Início" para mostrar apenas o horário de início e não o intervalo, assim ficando mais fácil de ordenar por horário outras tabelas.

Etapa de exclusão:

Código 7 - Exclusão de colunas (Emissora 1, 2 e NI).

```
#Retira as colunas das quais nao vamos usar
clean_merged_semana = new_merged_emissora.drop(['Total Domicílios |
Shr%','AB | Shr%', 'C1 | Shr%', 'C2 | Shr%', 'DE | Shr%', 'Masculino |
Shr%',
       'Feminino | Shr%', '4-11 anos | Shr%', '12-17 anos | Shr%',
       '18-24 anos | Shr%', '25-34 anos | Shr%', '35-49 anos | Shr%',
       '50-59 anos | Shr%', '60+ anos | Shr%', 'Total Indivíduos | Rch%',
       'AB | Rch%', 'C1 | Rch%', 'C2 | Rch%', 'DE | Rch%', 'Masculino | Rch%',
       'Feminino | Rch%', '4-11 anos | Rch%', '12-17 anos | Rch%',
       '18-24 anos | Rch%', '25-34 anos | Rch%', '35-49 anos | Rch%',
       '50-59 anos | Rch%', '60+ anos | Rch%', 'Total Indivíduos | Fid%',
       'AB | Fid%', 'C1 | Fid%', 'C2 | Fid%', 'DE | Fid%', 'Masculino | Fid%',
       'Feminino | Fid%', '4-11 anos | Fid%', '12-17 anos | Fid%',
       '18-24 anos | Fid%', '25-34 anos | Fid%', '35-49 anos | Fid%',
       '50-59 anos | Fid%', '60+ anos | Fid%', 'Unnamed: 0_y', 'Faixa Horária','Emissora
2','Emissora 3', 'Emissora 1', 'Emissora', 'Praça'], axis=1)
```

Fonte: do próprio autor (2022).

Deixamos apenas colunas das quais precisamos, assim fazendo como que possamos excluir as outras 20 colunas. Para isso usamos o comando .drop relacionando as colunas das quais queremos excluir. Deixando nossa tabela muito mais leve, precisa e apenas com as informações que iremos usar.

4.3.2. Como deve ser feita a agregação de registros e/ou derivação de novos atributos

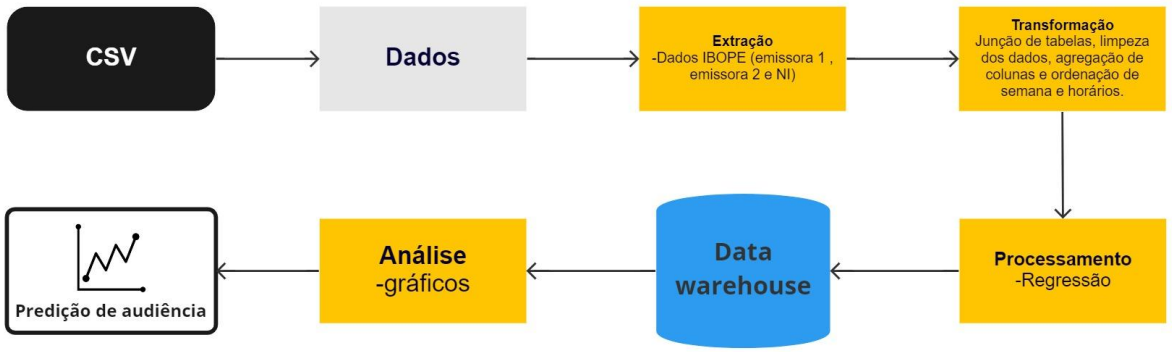
Para serem realizadas as derivações dos atributos na ferramenta, foi utilizada a função Split do Pandas, a qual é responsável por dividir uma string em pequenos pedaços utilizando um "separador", que pode ser escolhido no código.

As manipulações realizadas dos registros que foram disponibilizados pelo IBOPE, foram a divisão das colunas de "programa" e "categoria", que antes eram em conjunto, coluna "PROGRAMA/CATEGORIA", sendo assim ficando separadas, agregando à tabela apenas a sessão categoria. Além disso, em uma outra situação se viu necessária a fragmentação da data em três colunas diferentes, "ano", "mês" e "dia", o que antes era "Data" em apenas uma coluna, utilizando a mesma função.

Abaixo, segue um fluxograma básico, a fim de ilustrar as etapas da elaboração da nossa solução. Ela consiste em extrair da tabela em CSV, fornecida pelo cliente, os dados sobre os

últimos 2 anos da emissora. Após isso, ocorre a extração destes que a nós interessa (emissora 1, emissora 2 e NI), em sequência ocorre a transformação, que abrange a junção de tabelas, agregação de colunas, ordenação de semana e horários. A fase do processamento consiste na parte em que ocorrerá a regressão, que em seguida gerará dados que serão depositadas no Data warehouse, repositório central de informações que podem ser analisadas para tomar decisões mais adequadas, normalmente acessado por Analistas de negócios, engenheiros de dados, cientistas de dados e tomadores de decisões. Em próxima instância são analisados os dados, gerando os gráficos, que mostram de maneira mais simplificada o resultado da regressão.

Fluxograma 2 -



4.3.3. Identificação das features selecionadas, com descrição dos motivos de seleção.

Inicialmente, foi feita uma seleção das das features de acordo com o consenso do grupo, em relação às quais melhores cabiam para o modelo que seria utilizado. Pensando naquelas que mais agregavam e poderiam ter impacto no output da predição, optou-se por manter as features:

Tabela 1 - Features escolhidas.

Data
Hora de início
Dia da semana
Total Domicílios I Rat%
AB I Rat%
C1 I Rat%
C2 I Rat%
De I Rat%
Masculino I Rat %
4-11 anos I Rat%
12-17 anos I Rat%
18-24 anos I Rat%
25-34 anos I Rat%
35-49 anos I Rat%
50-59 anos I Rat%
60+ anos I Rat%
Programa
Categoria
Feriado
Ano
Dia
Mês

Fonte: do próprio autor (2022).

Sendo o Rat%, a medida média de domicílios que assistiram ao programa (Kantar IBOPE, 2016), acreditou-se para o começo dos testes que seria a porcentagem ideal para ser considerada, deixando assim de lado as medidas de alcance (Shr%) e fidelidade (Fid%).

Assim, considerou-se a relevância do espaço-tempo em que o programa está encaixado, uma vez que de acordo com o horário, dia, mês, ano, dia da semana e se esse é feriado ou não, há uma mudança de pontos atingidos e do perfil de telespectadores, podendo estabelecer uma correlação direta entre essas variáveis e o resultado final.

Além disso, observou-se que os segmentos que cada programa possuía afetava o desempenho no score de audiência das emissoras, juntamente com o perfil do público. Dessa forma foi considerado, cada um dos segmentos referentes às faixas horárias medidas. Ademais, após avaliações, tornou-se claro a relevância de dados sobre o público que estava assistindo, uma vez que esses se mostravam os que mais sofriam alterações de acordo com as demais variáveis.

Neste sentido, após a escolha das features, foi feita uma análise do impacto de cada variável a partir do conceito de multicolinearidade, ou seja, conjunto de características, considerando duas delas que carregam as mesmas ou similares informações, na qual possuem relação linear muito forte. Como consequência, em casos assim, o modelo pode acabar ignorando uma das características, tomando elas como redundantes.

Para conseguir obter os resultados, foram utilizadas as bibliotecas *numpy*, *pandas*, *seaborn* e *matplotlib.pylab* disponíveis para a linguagem Python, a partir do código abaixo:

Código - Código para gerar o mapa de multicolinearidade.

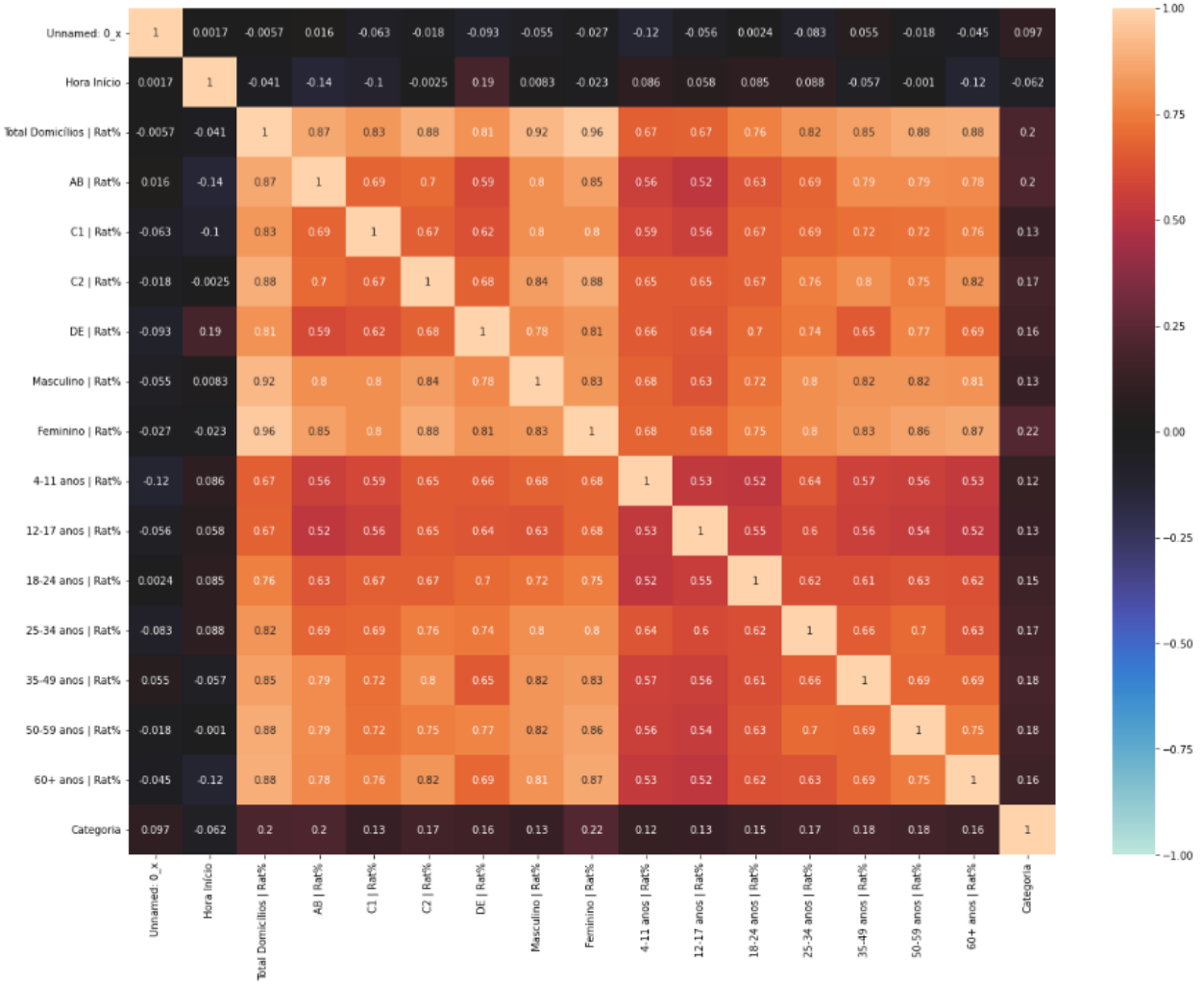
```
# Importação da biblioteca álgebra linear
import numpy as np
# Manipulação dos dados
import pandas as pd
# Visualização
import seaborn as sns
import matplotlib.pylab as plt
from matplotlib.pylab import rcParams

# Definindo tamanho da figura
rcParams['figure.figsize'] = 20, 15
# Matriz de correlação
matriz_correlacao = clean_merged.corr()
# Mapa de calor
sns.heatmap(matriz_de_correlacao, annot = True, vmin =-1, vmax=1, center=0)
# Definindo a posição dos tricks nos eixos
plt.yticks(rotation = 360)
plt.xticks(rotation = 90)
# Mostrando a figura
plt.show()
```

Fonte: do próprio autor (2022).

Dessa forma foi observado com a saída da execução, quais das features estariam com dados mais correlacionados, sendo as cores mais claras as associações mais similares, conforme observado no mapa abaixo.

Figura x - Mapa de multicolinearidade.



Fonte: do próprio autor (2022).

4.4. Modelagem

Para a Sprint 3, você deve descrever aqui os experimentos realizados com os modelos (treinamentos e testes) até o momento. Não deixe de usar equações, tabelas e gráficos de visualização de dados para melhor ilustrar seus experimentos e resultados.

Na modelagem dos dados nós utilizamos 3 modelos para o teste de predição, são eles: o KNN, Light GBM e a regressão linear, possuindo resultados diferentes em cada um deles.

No que tange a **regressão linear**, ela se apresenta como um modelo mais simples dentre todos os testados, por representar um modelo simples de machine learning para encontrar a reta que melhor explica a relação entre as features e um target contínuo, que, em linhas gerais, aproxima o target das features por uma reta (uma função linear). Desse modo, podemos fazer afirmações sobre valores não disponíveis no dataset. Então, para sumarizar o processo, nós iniciamos importando as tabelas e as bibliotecas necessárias para para a manipulação dos dados:

```
#importando bibliotecas
import matplotlib.pyplot as plt
import pandas as pd
import datetime as dt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import *
from google.colab import drive

#montando o drive
drive.mount('/content/drive')

#importando tabela completa emissora 1
data = pd.read_csv("/content/drive/Shareddrives/hefestos/CSV/Tabela_limpa_completa_float.csv", sep = ',')
data = data.drop(['Data','Unnamed: 0'], axis=1)
data[data['Feriado']== 1]
```

	Hora Início	Dia da Semana	Total Domicílios | Rat%	AB | Rat%	C1 | Rat%	C2 | Rat%	DE | Rat%	Masculino | Rat%	Feminino | Rat%	4-11 anos | Rat%	...	21	22	23	24	25	26	27	28	29	30
8064	6.000000	5	4.0	3.28	0.00	0.89	3.75	2.72	1.37	0.0	...	0	0	0	0	0	0	0	0	0	0
8065	6.083333	5	5.0	3.28	0.00	1.59	3.56	3.12	1.31	0.0	...	0	0	0	0	0	0	0	0	0	0
8066	6.166667	5	5.0	3.28	0.42	1.47	4.39	3.51	1.35	0.0	...	0	0	0	0	0	0	0	0	0	0
8067	6.250000	5	7.0	3.77	0.83	2.42	4.76	3.86	2.12	0.0	...	0	0	0	0	0	0	0	0	0	0
8068	6.333333	5	8.0	3.77	1.70	2.61	5.64	3.85	2.88	0.0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
215995	29.583333	1	5.0	1.43	0.71	2.70	4.21	1.82	2.42	0.0	...	0	0	0	0	0	0	0	0	0	0
215996	29.666667	1	5.0	1.51	0.71	2.70	4.21	1.82	2.48	0.0	...	0	0	0	0	0	0	0	0	0	0
215997	29.750000	1	5.0	1.51	0.14	3.06	4.21	1.77	2.48	0.0	...	0	0	0	0	0	0	0	0	0	0
215998	29.833333	1	6.0	1.51	0.35	4.10	4.21	2.28	2.61	0.0	...	0	0	0	0	0	0	0	0	0	0
215999	29.916667	1	6.0	1.51	0.58	4.10	4.63	2.45	2.70	0.0	...	0	0	0	0	0	0	0	0	0	0

8352 rows × 50 columns

Após isso, nós selecionamos quais seriam as variáveis utilizadas para o modelo preditivo e descartamos as outras e treinamos o modelo com o método “LinearRegression()” da biblioteca do Scikit-learn:

```
[ ] #dividindo x e y
    x = data.drop(['AB | Rat%','C1 | Rat%','C2 | Rat%','DE | Rat%','Masculino | Rat%','Feminino | Rat%','4-11 anos | Rat%',
    #y = data[['Total Domicílios | Rat%','AB | Rat%','C1 | Rat%','C2 | Rat%','DE | Rat%','Masculino | Rat%','Feminino | Ra
    y = data['Total Domicílios | Rat%']

    #dividindo treino e teste
    x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.15,  random_state = 100000)
```

```
[ ] #treinamento do modelo

    model = LinearRegression().fit(x_train ,y_train)
```

Verificação de qual o conjunto de teste:

```
x_test
```

	Hora Início	Dia da Semana	Feriado	Ano	Dia	Mes	1	2	3	4	...	21	22	23	24	25	26	27	28	29	30
142276	6.333333	6	0	2022	22	4	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
13220	27.666667	2	0	2020	3	8	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
95697	12.750000	4	0	2021	8	9	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
208439	23.916667	1	0	2021	17	10	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
200648	22.666667	1	0	2021	11	4	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200088	24.000000	1	0	2021	28	3	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
9856	11.333333	6	0	2020	17	7	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
167765	18.416667	7	0	2021	27	2	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
152757	15.750000	2	0	2022	13	6	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
17161	20.083333	6	0	2020	21	8	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

32832 rows × 36 columns

Então, fizemos um teste do modelo preditivo utilizando o método “model.predict()” da biblioteca do Scikit-learn, verificamos quais foram os resultados e computamos o R-dois com o método r2_score(y_test, y_pred) para verificar o quão acurado é o modelo:

```
[ ] y_pred = model.predict(x_test)
    y_pred
```

```
array([ 9.02855253,  7.29171739,  9.06295045, ..., 19.30343231,
        5.77169759, 19.46316708])
```

```
r2_score(y_test, y_pred)
```

```
0.4534275379102397
```

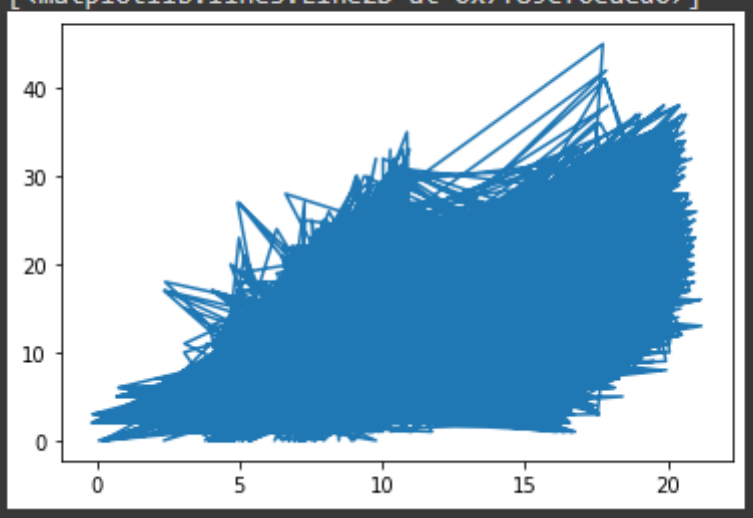
Como funciona o cálculo do R-dois:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

A seguir, o gráfico gerado pelo modelo preditivo:

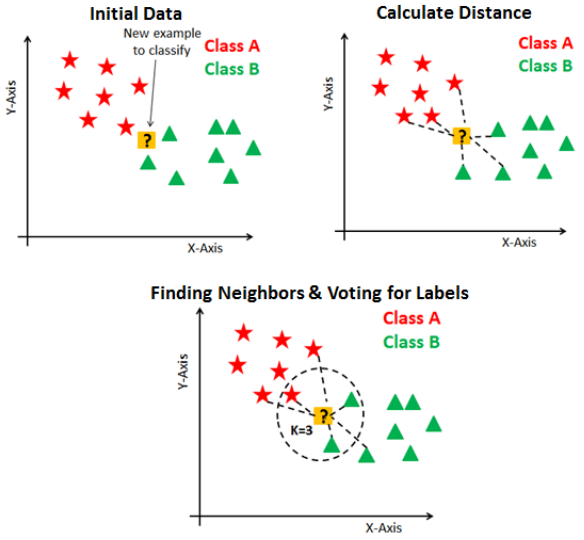
```
plt.plot(y_pred, y_test)
```

```
[<matplotlib.lines.Line2D at 0x7f89ef0eaed0>]
```



Com esse gráfico pode-se chegar à conclusão de que tal modelo teve pouca acurácia, uma vez que a formação do gráfico se distancia muito do formato de uma reta, que, no caso, seria o modelo ideal.

Já em outro teste realizado, utilizamos o modelo **K-Nearest Neighbors (KNN)**, ou métodos vizinhos mais próximos, que consiste em uma ferramenta de machine learning supervisionado, que utiliza ferramentas matemáticas para comparar dados semelhantes entre si com o objetivo de inferir uma determinada característica sobre um data point desconhecido.



Então, para expor o processo, iniciamos importando as tabelas e as bibliotecas necessárias para para a manipulação dos dados:

```
#------------------------ IMPORTAÇÃO DE TABELAS ---------------------------
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
dataset = pd.read_csv('/content/drive/Shareddrives/hefestos/CSV/Tabela_limpa_completa.csv')
dataset.head()
dataset.columns
```

```
Index(['Unnamed: 0', 'Unnamed: 0.1', 'Unnamed: 0_x', 'Data', 'Hora Início',
       'Dia da Semana', 'Total Domicílios | Rat%', 'AB | Rat%', 'C1 | Rat%',
       'C2 | Rat%', 'DE | Rat%', 'Masculino | Rat%', 'Feminino | Rat%',
       '4-11 anos | Rat%', '12-17 anos | Rat%', '18-24 anos | Rat%',
       '25-34 anos | Rat%', '35-49 anos | Rat%', '50-59 anos | Rat%',
       '60+ anos | Rat%', 'Categoria', 'Ano', 'Dia', 'Mes', 'Feriado', '1',
       '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14',
       '15', '16', '17', '18', '19', '20', '21', '22', '23', '24', '25', '26',
       '27', '28', '29', '30'],
      dtype='object')
```

Após isso, separamos quais os outputs desejados:

```
[ ] df_both = dataset.sample(len(dataset))
    #remoção do output desejado
    df_both.drop(['Total Domicílios | Rat%'], axis=1, inplace=True)
    df_both.drop(['Masculino | Rat%'], axis=1, inplace=True)
    df_both.drop(['Feminino | Rat%'], axis=1, inplace=True)
    df_both.drop(['AB | Rat%'], axis=1, inplace=True)
    df_both.drop(['C1 | Rat%'], axis=1, inplace=True)
    df_both.drop(['C2 | Rat%'], axis=1, inplace=True)
    df_both.drop(['DE | Rat%'], axis=1, inplace=True)
    df_both.drop(['4-11 anos | Rat%'], axis=1, inplace=True)
    df_both.drop(['12-17 anos | Rat%'], axis=1, inplace=True)
    df_both.drop(['18-24 anos | Rat%'], axis=1, inplace=True)
    df_both.drop(['25-34 anos | Rat%'], axis=1, inplace=True)
    df_both.drop(['35-49 anos | Rat%'], axis=1, inplace=True)
    df_both.drop(['50-59 anos | Rat%'], axis=1, inplace=True)
    df_both.drop(['60+ anos | Rat%'], axis=1, inplace=True)
    df_both.drop(['Data'], axis=1, inplace=True)
    df_both.columns
```

```
Index(['Unnamed: 0', 'Unnamed: 0.1', 'Unnamed: 0_x', 'Hora Início',
       'Dia da Semana', 'Categoria', 'Ano', 'Dia', 'Mes', 'Feriado', '1', '2',
       '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15',
       '16', '17', '18', '19', '20', '21', '22', '23', '24', '25', '26', '27',
       '28', '29', '30'],
      dtype='object')
```

Então, realizamos a divisão dos eixos do modelo e separamos os grupos de treino e de teste. Rodamos então o algoritmo através do método KNeighborsClassifier() e verificamos a acuracidade do modelo, a qual foi muito baixa no treino (25% de acuracidade) e no teste (5% de acuracidade).

```
[ ] y = dataset['Total Domicílios | Rat%']
    X_train = df_both
    X_train, X_test, y_train, y_test = train_test_split(X_train, y, test_size=0.33, random_state=1000)
```

```
[ ] # Instanciação do obj Algoritmo
    knn = KNeighborsClassifier(n_neighbors=7)
    # Treino # x = Features, y = Label/Target
    knn.fit(X_train, y_train.squeeze()) # squeeze() -> df para series
    # Teste de Acuracidade (accuracy)
    print('Acuracidade (treino): ', knn.score( X_train, y_train ))
    print('Acuracidade (teste): ', knn.score( X_test, y_test ))
```

```
Acuracidade (treino):  0.25603311307953003
Acuracidade (teste):  0.05846520192161261
```

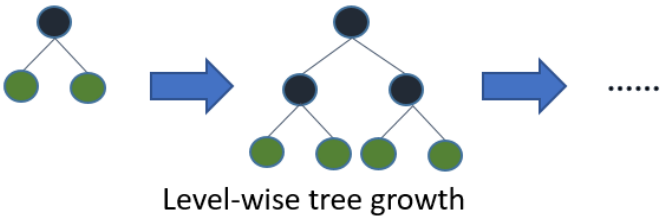
```
[ ] # realizando predições com o conjunto de teste
    y_pred = knn.predict(X_test)
    y_pred
```

```
array([3, 7, 2, ..., 1, 9, 2])
```

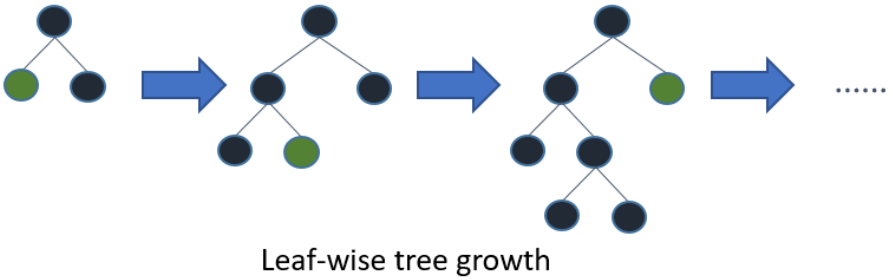
Em nosso teste com melhor resultado, foi utilizado o **LightGBM**, que consiste em um método baseado em gradient boosting, o qual é comumente utilizado para modelos de regressão e classificação. Ela constrói o modelo em etapas, como outros métodos de boosting, e os generaliza, permitindo a otimização de uma função de perda diferenciável arbitrária.

LightGBM é um sistema o qual distribui a estrutura de aumento de gradiente usando uma estrutura de aprendizagem baseada em árvore, sendo assim, baseado em histograma e coloca valores contínuos em compartimentos discretos, o que leva a um treinamento mais rápido e um uso mais eficiente da memória.

A estrutura usa um algoritmo de crescimento de árvore em folha que é diferente de muitos outros algoritmos baseados em árvore, os quais usam crescimento em profundidade. Os algoritmos de crescimento de árvores em folhas tendem a convergir mais rapidamente que os em profundidade. No entanto, eles tendem a ser mais propensos a sobreajuste, como pode ser percebido no gráfico abaixo a diferença entre as duas supracitadas:



Crescimento de árvore em nível



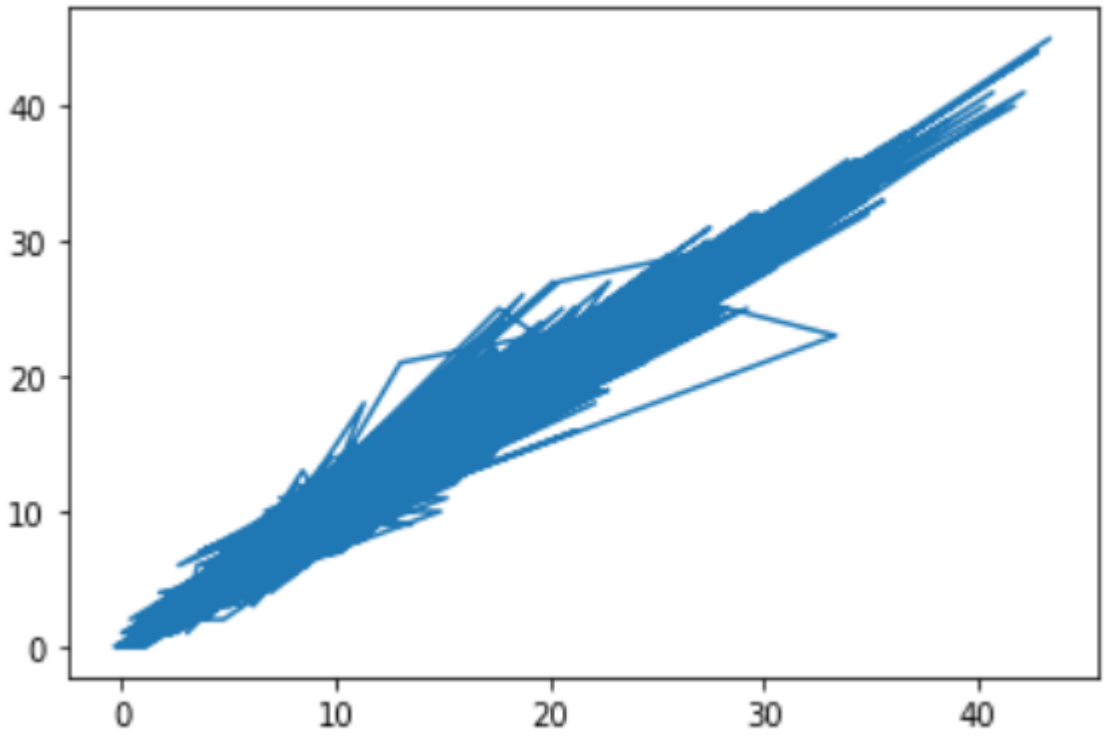
Crescimento de árvore em folha

Com isso, foram obtidos os seguintes resultados:

Para a Sprint 4, você deve realizar a descrição final dos experimentos realizados (treinamentos e testes), comparando modelos. Não deixe de usar equações, tabelas e gráficos de visualização de dados para melhor ilustrar seus experimentos e resultados.

4.5. Avaliação

Para o projeto, após a análise do comportamento de cada modelo testado, o grupo chegou à conclusão de que o melhor modelo a ser utilizado seria o LightGBM, devido ao seu alto desempenho durante o teste com várias métricas. Isso demonstra que tal modelo é o que consegue chegar aos valores mais próximos aos que correspondem à realidade. Tal fato pode ser visto através do gráfico gerado pelo modelo, o qual se assemelha muito a uma reta:



Para a avaliação do modelo, nós utilizamos diversos métodos de validação sendo eles o **R-dois**, a **acurácia**, o **erro médio absoluto** e o **erro médio quadrático**, que se mostraram como os mais promissores. Começando pelo R-dois, que já teve sua fórmula descrita no documento, nós obtivemos tal resultado:

```
#R²
from sklearn import metrics
from sklearn.metrics import r2_score
print(metrics.r2_score(y_test, y_pred_test))
```

```
0.9849967515110364
```

A fórmula para o cálculo do R-dois:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Já quanto à **acurácia**, que é uma comparação percentual entre os valores reais e os valores encontrados, nós realizamos o teste e encontramos os seguintes resultados:

```
[ ]  #Acurácia do treino
     train_score = gbm.score(X_train,y_train)
     print(train_score)

     0.9960349830911998

[ ]  #Acurácia do teste
     test_score = gbm.score(X_test,y_test)
     print(test_score)

     0.9849967515110364
```

Para o cálculo do **erro médio absoluto**, nós utilizamos o método "mean_absolute_error" para a verificação da qualidade do modelo, o qual precisa apresentar valores mais baixos para que o modelo seja considerado bom:

```
[ ]  #erro médio absoluto
     from sklearn.metrics import mean_absolute_error
     mean_absolute_error(y_train, y_pred_train)

     0.310854725614027
```

Fórmula para o cálculo do erro médio absoluto:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | Yi - \hat{Y}i |$$

Para o cálculo do **erro médio quadrático**, nós utilizamos o método "mean_squared_error" para a verificação da qualidade do modelo, o qual, assim como no caso do erro médio absoluto, precisa apresentar valores mais baixos para que se afirme uma qualidade maior no modelo:

```
[ ]  #erro médio quadrático
     from sklearn.metrics import mean_squared_error
     mean_squared_error(y_test, y_pred_test)

     0.6318695743936463
```

Fórmula do erro médio quadrático:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Nesta seção, descreva a solução final de modelo preditivo, e justifique a escolha. Alinhe sua justificativa com a seção 4.1, resgatando o entendimento do negócio e explicando de que formas seu modelo atende os requisitos. Não deixe de usar equações, tabelas e gráficos de visualização de dados para melhor ilustrar seus argumentos.

4.6 Comparação de Modelos

5. Conclusões e Recomendações

Escreva, de forma resumida, sobre os principais resultados do seu projeto e faça recomendações formais ao seu parceiro de negócios em relação ao uso desse modelo. Você pode aproveitar este espaço para comentar sobre possíveis materiais extras, como um manual de usuário mais detalhado na seção “Anexos”.

Não se esqueça também das pessoas que serão potencialmente afetadas pelas decisões do modelo preditivo, e elabore recomendações que ajudem seu parceiro a tratá-las de maneira estratégica e ética.

6. Referências

Nesta seção você deve incluir as principais referências de seu projeto, para que seu parceiro possa consultar caso ele se interessar em aprofundar.

Utilize a norma ABNT NBR 6023 para regras específicas de referências. Um exemplo de referência de livro:

LUCK, Heloisa. **Liderança em gestão escolar**. 4. ed. Petrópolis: Vozes, 2010.

SOBRENOME, Nome. **Título do livro**: subtítulo do livro. Edição. Cidade de publicação: Nome da editora, Ano de publicação.

MEDIA, Kantar IBOPE. **Dados da audiência nas 15 praças praças regulares com base no ranking consolidado – 04/07 a 10/07.** Disponível em: ‹https://www.kantaribopemedia.com/dados-de-audiencia-nas-15-pracas-regulares-com-base-no-ranking-consolidado-0407-a-1007/›. Acesso em: 21 de agosto de 2022.

Anexos

Utilize esta seção para anexar materiais como manuais de usuário, documentos complementares que ficaram grandes e não couberam no corpo do texto etc.