O Oráculo
TV Gazeta

inteli
instituto
de tecnologia
e liderança

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
11/08/2022	Yasmin	1.0	1. Introdução
11/08/2022	Gabrio e Patricia	1.0	4.1.2. Matriz SWOT e 4.1.5. Matriz de Riscos
11/08/2022	Gustavo	1.0	4.1.1. Contexto da indústria
11/08/2022	Gabrio, Gustavo, Patricia, Tainara, Yasmin e Victor	1.0	4.1.3. Planejamento Geral da Solução
11/08/2022	Gustavo e Tainara	1.0	4.1.4. Value Proposition Canvas
11/08/2022	Gabrio, Gustavo, Patricia, Tainara, Yasmin e Victor	1.0	4.2. Compreensão dos Dados
12/08/2022	Yasmin	1.1	6. Referências
15/08/2022	Gabrio, Gustavo, Yasmin, Tainara, Victor, Patricia	2.0	4.1.6. Personas
17/08/2022	Gabrio, Yasmin, Patricia	2.0	4.1.7 Jornada do Usuário
26/08/2022	Tainara e Victor	2.1	4.3.2 Manipulação de dados e registros
26/08/2022	Gustavo e Patricia	2.1	4.3.2 Agregação de Registros e derivação de novos atributos
26/08/2022	Gustavo, Tainara e Yasmin	2.1	4.3.4 Remoção e substituição de valores ausentes, em branco, ou desconsiderados
26/08/2022	Gustavo, Tainara e Yasmin	2.1	4.3.5 Identificação das features selecionadas
27/08/2022	Gábrio, Patricia	2.1	4.3.1 Anonimização dos dados
28/08/2022	Gábrio	2.2	Revisão, padronização e formatação do documento.
11/09/2022	Yasmin	3.0	4.4. Modelagem - Modelo de regressão linear
11/09/2022	Victor	3.0	4.4. Modelagem - Árvore de decisão - Modelo de regressão

11/09/2022	Patricia e Gustavo	3.0	4.4. Modelagem - KNN - Modelo de regressão
11/09/2022	Gabrio	3.0	4.4 - Modelagem - RFE, Modelo de LightGBM e avaliação do modelo
11/09/2022	Tainara	3.0	4.4 - Modelagem - Random Forest - Modelo de regressão
25/09/2022			
25/09/2022			
25/09/2022			
25/09/2022			
25/09/2022			

Sumário

1. Introdução

Apresente de forma sucinta o parceiro de negócio, seu porte, local, área de atuação e posicionamento no mercado. Maiores detalhes deverão ser descritos na seção 4

Descreva resumidamente o problema a ser resolvido (sem ainda mencionar a solução).

Caso utilize citações ao longo desse documento, consulte a norma ABNT NBR 10520. É sugerido o uso do sistema autor-data para citações.

A medição de audiência na televisão acontece por meio da tradicional amostragem com o aparelho chamado People Meter em um lar escolhido, este possui mapas de calor para identificar se a pessoa está acompanhando determinada mídia e quem participa da pesquisa tem seu próprio número, informando o sexo, idade, classe economica e programação assistida. Embora o PeopleMeter tenha o principal objetivo de medir o índice de audiência, verificando o tamanho e a composição do público que acompanha determinada programação, por meio desse método, as informações são insuficientes para a medição do sucesso dos programas de TV, tal como sua classificação antecipada e se o público atenderá as expectativas da audiência da programação, ou seja, em razão das limitações da tecnologia existente, há dificuldade no entendimento e previsão do comportamento da audiência, tendo grandes chances de gerar prejuízos financeiros, mobilização frequente de profissionais e equipes devido a atualizações e cancelamentos da programação, qualidade da programação e, claro, a perda de espectadores.

Desse modo, não há uma visão 360° para isolar os vários fatores que levam ao declínio de audiência, dado que, todos os dias os espectadores são inundados com diversas opções de entretenimento, havendo uma necessidade de informações do que está funcionando e a razão que faz o espectador não assistir determinada programação ou mudar de canal, preferindo a concorrência. A exibição da programação com conteúdos selecionados de acordo com o que interessa quem mais importa, o público, são escalados sem uma base preditiva consistente nas decisões sobre produção e cronogramas de programas, alterando a classificação e visão do canal, principalmente ao se tratar da atualidade, em que há intensa competição entre canais devido o surgimento de novas mídias.

2. Objetivos e Justificativa

2.1. Objetivos

Descreva resumidamente os objetivos gerais e específicos do seu parceiro de negócios

2.2. Justificativa

Faça uma breve defesa de sua proposta de solução, escreva sobre seus potenciais, seus benefícios e como ela se diferencia.

3. Metodologia

Descreva as etapas metodológicas que foram utilizadas para o desenvolvimento, citando o referencial teórico. Você deve apenas enunciar os métodos, sem dizer ainda como ele foi aplicado e quais resultados obtidos.

3.1. CRISP-DM

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

3.2. Ferramentas

Descreva brevemente as ferramentas utilizadas e seus papéis (Google Colaboratory)

3.3. Principais técnicas empregadas

Descreva brevemente as principais técnicas empregadas, algoritmos e seus benefícios

4. Desenvolvimento e Resultados

4.1. Compreensão do Problema

4.1.1. Contexto da indústria

A TV aberta possui um mercado que conta com diversos players com atuação intensa, entre os principais pode-se citar: Rede Globo, o maior conglomerado de mídia e comunicação da américa latina e a segunda maior do mundo, a Record, considerada a segunda maior emissora do Brasil e a quinta maior do mundo, o SBT, uma emissora fundada pelo empresário e animador Silvio Santos e por fim a Bandeirantes, fundada pelo empresário João Jorge Saad e sendo considerada a quarta maior rede de televisão do país.

As emissoras abertas oferecem para o público uma variedade de programas de forma gratuita. O foco das emissoras que possuem esse modelo de negócio é capturar a atenção dos telespectadores para vendê-la aos anunciantes e assim gerar rentabilidade. Consequentemente, os produtos desenvolvidos tendem a ser genéricos buscando alcançar uma alta taxa de aceitabilidade entre o maior número de telespectadores.

A atual infraestrutura tecnológica provoca um gargalo para a inovação da TV aberta. Uma das primeiras estratégias utilizadas foi a preservação da qualidade de imagem superior às outras mídias. Entretanto, com o aumento da qualidade em dispositivos, redes sociais e serviços de streaming concorrentes, a rede televisiva se viu frente a um problema em que se seria necessário buscar novas estratégias. A convergência de diversos serviços para uma única plataforma digital é um movimento atual das grandes emissoras e teve um efeito de encadeamento, sendo adotado pelos grandes players. O público mais jovem consome conteúdos mais curtos e sucintos, sendo essa a tendência presente no mercado midiático. Portanto, a televisão deve expandir o seu alcance para uma diversidade de aparelhos maior e levar em consideração o contexto do consumo, colocando em cartaz conteúdos com diferentes tipos de duração.

Para abordar uma melhor percepção do contexto do mercado, foi analisado o cenário de atuação da empresa parceira. A ferramenta escolhida foram as 5 forças de porter.

Poder de negociação dos fornecedores: Fornecedores são quem provêm insumos para criação de produtos. O produto da TV aberta são os programas. Esses programas podem ter autoria própria ou ter a licença contratada de outra empresa. Classificam-se em 2 categorias, nacionais e internacionais. No âmbito nacional o poder de barganha dos fornecedores é baixo, pois o público alvo se encontra na mídia tradicional e no Brasil existem poucas opções e uma

discrepância grande de audiência. Os internacionais possuem um poder de barganha maior pois além de atenderem ao mundo todo, também atendem diferentes tipos de mídia e plataformas.

Poder de negociação dos clientes: Cliente é quem gera rentabilidade para a empresa. No caso das televisões abertas, quem gera essa rentabilidade são os anunciantes. A televisão atrai a atenção dos telespectadores com os programas e durante intervalos propaga os anúncios. Na atualidade, devido a globalização e ao acesso facilitado a fontes de informação e entretenimento, existem diversas alternativas para empresas que desejam propagar seus produtos, que vão de anúncios em sites e vídeos. Também existem produtos digitais que não possuem em seu modelo de negócio o conceito de anunciantes. Apesar disso, pesquisas indicam que as pessoas preferem assistir anúncios e pagar mais barato ao invés de pagar mais caro por um produto sem anúncios, o que pode acarretar uma mudança de proposta dos substitutos. Em suma, existe uma competitividade quente no mercado e os clientes possuem uma gama de opções muito grande.

Ameaça de produtos substitutos: É importante primeiro analisar a "big picture" dos produtos oferecidos pela grande mídia. De forma geral, a intenção é gerar entretenimento para a audiência. Com a chegada dos computadores e principalmente da recente "geração smartphone" as pessoas têm um acesso muito maior a diferentes fontes de entretenimento e até mesmo nascem conectadas. Está na ponta do dedo, literalmente. É de se observar a modificação no consumo que mais apetece o público. De um tempo pra cá, a preferência é por vídeos curtos e rápidos. Em 2021, aconteceu um boom de um aplicativo que possui essa proposta, ele conseguiu se manter no topo e brigou com gigantes. Isso forçou a adaptação da mídia digital, fazendo com que os concorrentes entrassem nessa tendência. Conclui-se que a ameaça é grande, pois além de ser um cardápio vasto, novas tendências podem surgir e mudar o esquema do jogo.

Ameaça de entrada de novos concorrentes: O mercado é um mar vermelho de oportunidades. A entrada de concorrentes na tv aberta é extremamente dificultosa pois as empresas que possuem relevância tem longevidade e procuram a identificação com os consumidores, especialmente filiais regionais, que apelam para a empatia e buscam a fidelidade, com sucesso.

Rivalidade entre os concorrentes: As marcas já estão consolidadas e possuem um público fiel, podendo existir até mesmo uma relação de amor ou ódio entre o público. Há uma diferença muito grande de audiência entre as competidoras, análogo a um monopólio. Entre as concorrentes existe uma tensão sobre o câmbio de funcionários entre os grandes players (Globo x Record, por exemplo), pois são considerados representantes da marca e carregam um poder de publicidade com eles. De um tempo pra cá, houve uma flexibilização dando uma maior liberdade aos artistas, mas ainda não é uma prática comum.

4.1.2. Análise SWOT

MATRIZ SWOT – FOFA		
	Fatores Positivos	**Fatores Negativos**
Fatores Internos	**Forças** - Marca consolidada e com alta reputação no mercado; - Profissionais capacitados; - Conhecimento do segmento; - Representante de uma rede de TV líder no Brasil; - Aquisições de tecnologias que permitam a expansão da emissora. - Audiência e alcance; - Time de Marketing Forte.	**Fraquezas** - Tradicionalismo; - Sistema obsoleto e sem precisão de dados; - Inflexibilidade na programação devido à contratos; - Time de inovação pequeno; - Dependência de outros agentes para análise de audiência.
Fatores Externos	**Oportunidades** - Expansão da programação; - Espaço exclusivo para comerciais de empresas externas na emissora; - Liderança local entre as emissoras concorrentes; - Exclusividade na trasmissão de grandes eventos (copa do mundo, brasileirão, olimpíadas); - Criação de uma tecnologia de predição de audiência para a emissora criar novas estratégias visando alcançar um maior público.	**Ameaças** - Serviços de streaming virtuais; - Redes sociais, smartphones e internet; - Aderência dos telespectadores aos programas do catálogo que se alteram por períodos de tempo; - Maturidade do mapeamento da programação dos concorrentes.

Figura 1 - Matriz SWOT da TV Gazeta (Fonte: Criação própria).

4.1.3. Planejamento Geral da Solução

a) quais os dados disponíveis (fonte e conteúdo - exemplo: dados da área de Compras da empresa descrevendo seus fornecedores)

Os dados disponíveis apresentam fontes do Kantar Ibope Media(Kantar IBOPE Media -), proveniente do Kantar Media, líder no mercado de pesquisa de mídia da América Latina, que disponibiliza dados para a tomada de decisão de clientes. Em relação ao conteúdo, este contém informações que informam ao cliente o valor da audiência(Rat%), fidelidade(Fid%), "share"(Shr%) e "reach"(Rch%) dentro de um determinado período (data e hora de início) para determinados públicos (faixa etária, classe social e sexo).

b) qual a solução proposta (pode ser um resumo do texto da seção 2.2)

A solução a ser desenvolvida se baseia na alimentação de sistemas de machine learning, a partir de dados do IBOPE, utilizando recursos de modelagem preditiva para medir com precisão

a estimativa de audiência da faixa-horária do canal comparando com score passados, assim como a composição do público espectador que acompanha a programação exibida e suas preferências previstas. A implementação de tal sistema possibilita o retorno de padrões e peso de atributos existentes nos dados tabulados com o score da audiência, de modo que sirva como auxílio nas decisões das produções e cronogramas de programas. Este é um método algoritmo capaz de usar dados inputados e prever o alcance do público potencial em diferentes cenários de distribuição utilizando variáveis, o qual considera o gênero do programa, data semanal, tempo da transmissão (faixa-horária), sexo e classe social, fornecendo padrões de aumento e queda de conteúdos.

A solução é treinável e melhora iterativamente a fim de gerar novas métricas de desempenho com visão ampla do atendimento de necessidades e das decisões estratégicas a serem tomadas para maximizar o retorno de investimentos em programação. Desse modo, é possível reagir com maior flexibilidade e exatidão às mudanças imprevistas, desempenhando ações antecipadas ao evento, quanto à estreia de programas, análise de resultados e a tração de planos de reversão de programas que não tiveram uma audiência tão boa, além de entender quais variáveis estão influenciando com que o produto esteja indo bem ou ruim e o que garante o alcance e a fidelidade do público.

c) qual o tipo de tarefa (regressão ou classificação)

Considerando a necessidade do projeto, de estimar possíveis valores de audiência a partir de uma ou múltiplas entradas(inputs), e os pesos de cada variável nesta predição, é reconhecida a demanda da implementação de métodos de regressão linear. Isso se dá pelo fato de que os valores resultantes esperados, como score de audiência, peso de atributos e outros, devem ser de característica contínua, numérica. Além disso, para inferir a influência destes atributos na predição final, é preciso entender a relação entre variáveis independentes e a variável de saída (output). Desta forma, é possível explicar a preferência pelo modelo regressivo, o qual tem por característica e finalidade, as próprias exigências mencionadas acima.

d) como a solução proposta deverá ser utilizada

O sistema preditivo desenvolvido deve ser utilizado para gerar métricas de dados acerca do desempenho da audiência, levando em consideração as variáveis desejáveis delimitadas preliminarmente, para assim, possibilitar a criação de estratégias internas que atendam as necessidades da empresa.

Esse sistema deverá ser continuamente alimentado com dados do IBOPE para realização da conversão desses dados para as métricas de estimativa de audiência das faixas-horárias do canal, uma vez que o modelo preditivo é treinável e necessita de inputs de dados para obtenção de uma maior maturidade da predição fornecida. O acesso do sistema por usuários será feito via plataforma Google Colab, a qual apresentará uma organização visual mostrando a segmentação de variáveis que consideram diferentes cenários. Por meio de campos de formulários que utilizarão filtros, o usuário terá acesso a uma estrutura de controle para selecionar e executar opções que irão coletar informações do sistema. Os resultados dos parâmetros gerados serão mostrados em formatos de gráficos.

Posteriormente, o usuário poderá utilizar os resultados obtidos através do sistema de predição para elaborar estratégias que visem impactar a programação da emissora e a venda de espaços publicitários para empresas externas, corroborando para um aumento significativo de lucros e alcance da marca.

e) quais os benefícios trazidos pela solução proposta

- análise de diferentes perspectivas (em relação ao desempenho de programas e
- emissoras);
- visão mais ampla de como a audiência se comporta entre os diferentes canais;
- exploração dos principais fatores que afetam a audiência;
- diminuição de implicações financeiras e maior retorno de investimentos;
- estimativa instantânea dos resultados;
- melhor seleção de conteúdos considerados atraentes pelo público;
- elaboração de estratégias que visam o aumento da popularidade;
- decisão baseada em dados mais precisos.

f) qual será o critério de sucesso e qual medida será utilizada para o avaliar

O desempenho do sistema preditivo poderá ser avaliado através do modelo desenvolvido que apresentará a emissora diferentes cenários possíveis para sua audiência baseado em métricas definidas que permitirão criação de estratégias assertivas para impulsionamento da marca. Além disso, com relação a venda de espaços publicitários ao longo da programação, o sistema poderá gerar um maior retorno financeiro para a emissora, visto a possibilidade de criação de nichos de alcance de audiência para públicos específicos, atendendo o objetivo de comerciais de diferentes produtos ou serviços.

4.1.4. Value Proposition Canvas

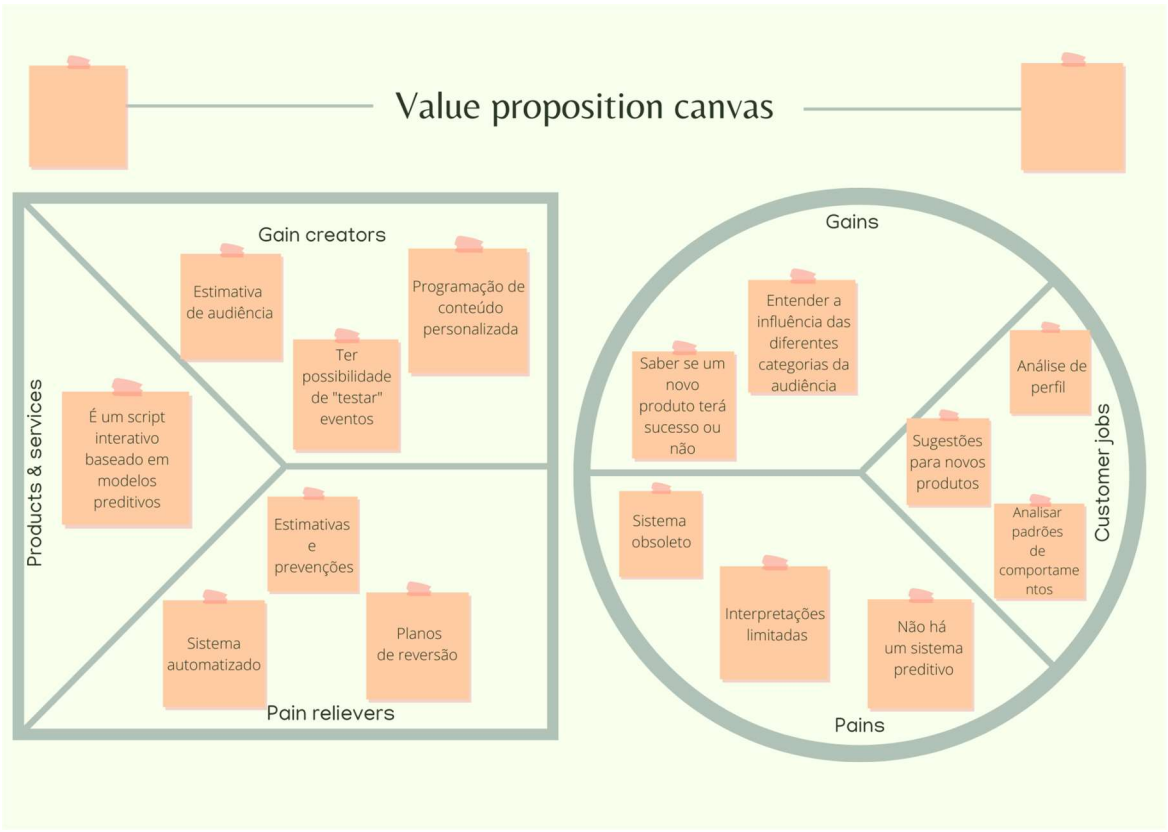


Figura 2 - Value proposition canvas (Fonte: Criação própria).

4.1.5. Matriz de Riscos

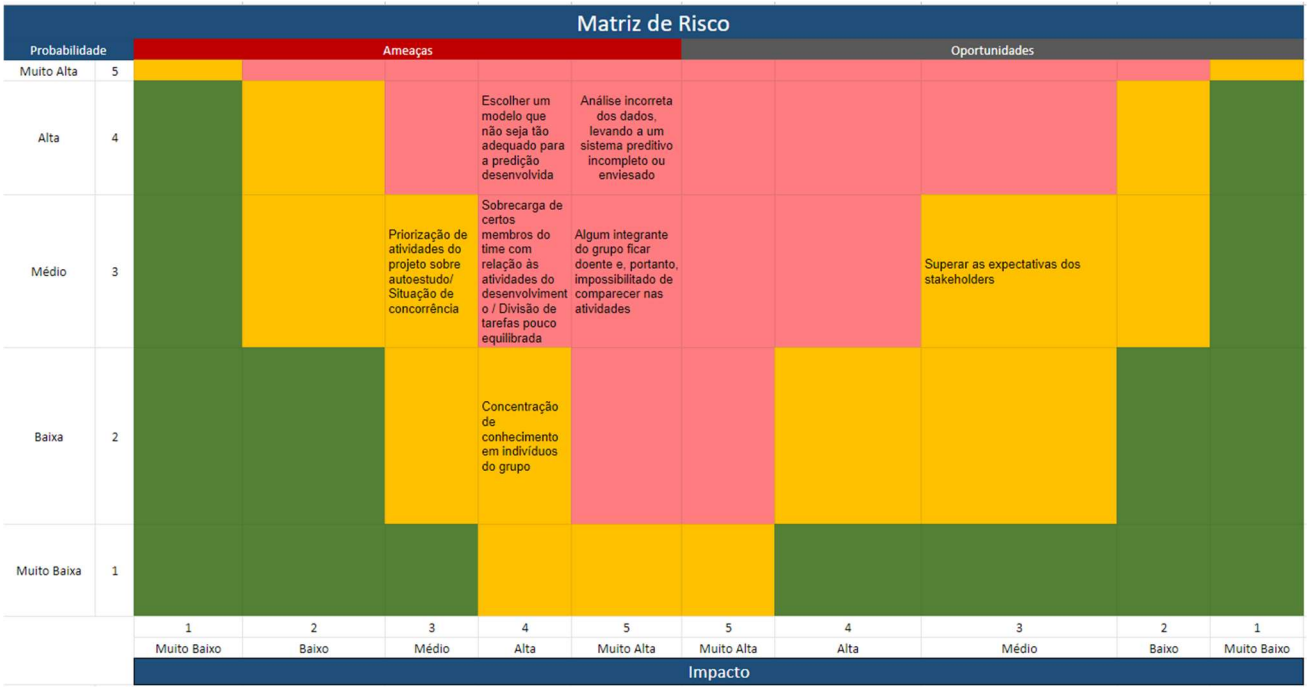


Figura 2 - Matriz de risco desenvolvida pelo grupo (Fonte: Criação própria).

4.1.6. Personas

Persona - *Representante da TV Gazeta*

Nome: Thiago Silva Schneider

"Doravante, de agora em diante; em direção ao futuro."

Idade: 37 anos

Ocupação: Gerente de Operação e Programação

Biografia:

- Nasceu em Vitória, ES;
- Graduado em Engenharia da Computação pela UFES;
- Mestrado em Business Analytics pela UFRGS;
- 5 anos de atuação no mercado de Data science;
- Com 7 anos de experiência na TV Gazeta com programação e operações, começou como coordenador de programação de TV e hoje exerce o cargo de gerente de programação e operações.

Interesse:

- Ter a possibilidade de ter um menor custo de gastos com armazenamento de dados;
- Compreender o impacto e participação de certos atributos na taxa de audiência final;
- Ter uma publicidade mais direcionada a públicos e nichos específicos.

Motivações:

- Precisão de alcance de audiência de um novo programa;
- Atrair um maior público para a emissora;
- O processo que é atualmente feito é impreciso e manual.

Dores:

- Alto Gasto com armazenamento de dados;
- Imprevisibilidade do sistema*;
- Ausência de um sistema automatizado;
- Falta de assertividade do sistema;
- Falta de agilidade na geração de resultados

4.1.7. Jornadas do Usuário

Thiago Silva Schneider

Cenário: Thiago precisa compreender fatores críticos do público em determinada faixa horária para inserção de um evento, seja ele um novo programa ou comercial.

Expectativas

Espera-se que o tempo excessivo na preparação de dados seja utilizado em insights, obtendo assertividade na predição da taxa de audiência para novas programações e redução de gastos.

FASE 1 (Gatilho e coleta dos dados)	**FASE 2** (Input dos dados)	**FASE 3** (Predição)	**FASE 4** (Incertezas)	**FASE 5** (Decisão final)
1. Recebimento de um novo programa; 2. Criação de estratégias de desempenho para os programas da rede;	1. Identificar quais dados serão utilizados; 2. Inserir tabela com os dados no sistema (fatores críticos que serão analisados)	1. Aguarda resultado da predição; 2. Recebe e Visualiza resultados; 3. Analisa as informações fornecidas pelo script do colab; 4. Filtra resultados para parâmetros específicos.	1. Identifica a audiência existente para programas similares na mesma faixa-horária; 2. Verifica se os parâmetros obtidos estão similares a predições anteriores. 3. Se questiona sobre a acuracidade da predição	1. Utiliza o sistema para compreender a audiência baseada em fatores críticos; 2. Cria estratégias para aprimorar o desempenho de programas exibidos ou futuros, utilizando os resultados do sistema; 3. Realiza venda de espaços publicitários com base no resultado do sistema. 4. Direciona recursos para maximizar o impacto de novos programas, de acordo com o desempenho analisado
Quando a dependência para obter dados diminuirá?	Poderia existir uma maneira mais automatizada ou fácil de inserir os dados?	Será que eu devo confiar nessa ferramenta?	Será que devo procurar outros meios para confimar o resultado?	Com esse sistema preditivo consigo...

Oportunidades

- Estruturação de um time de datascience, para a manutenção e atualizações do modelo, a fim de torná-lo mais eficiente e atual.
- Criação de interface Web para facilitar o uso do script
- Comparar a audiência estimada com outras emissoras para um mesmo horário

Responsabilidades

Atribua responsabilidades a pessoas/equipes e respectivas ações para resolver problemas e alcançar as melhorias

- Responsabilidades do time de marketing de gerenciar os recursos publicitários frente aos dados obtidos pela ferramenta
- RH para a contratação de novos desenvolvedores e cientistas de dados
- Responsabilidade dos cientistas de dados de constantemente alimentarem o modelo
- Equipe de UX para desenvolver o conceito de uma nova plataforma para a ferramenta.

Figura 3 - Jornada do usuário (Fonte: Criação própria)

4.2. Compreensão dos Dados

1. **Descreva os dados a serem utilizados (disponibilizados pelo cliente e outros se tiverem sido incluídos), detalhando a fonte, o formato (CSV, XLSX, banco de dados, etc.), o conteúdo e o tamanho.**

A fonte dos dados disponíveis vem do Kantar Ibope Media(Kantar IBOPE Media -), proveniente do Kantar Media. Enquanto o formato do arquivo dos dados é XLSX, ou seja, Planilha do Microsoft Office Excel.

Além disso, o conteúdo desse arquivo contém 18 abas, cada aba referencia uma emissora e um período da semana. Sendo 3 delas referenciando a emissora Total-Ligados-Especial("TLE - Seg a Sex", "TLE - Sáb" e "TLE - Dom"), outras 3 referenciando a emissora Principal, do parceiro("Emissora Principal (xxx) - Seg a Sex", "Emissora Principal (xxx) - Sab" e "Emissora Principal (xxx) - Dom"), mais 3 referenciam a emissora concorrente A ("Emissora concorrente A (yyy) - Seg a Sex", "Emissora Concorrente A(yyy) - Sab" e "Emissora concorrente A (xxx) - Dom"), mais 3 referenciando a emissora concorrente B("Emissora concorrente B (zzz) - Seg a Sex", "Emissora concorrente B (zzz) - Sab" e "Emissora concorrente B (zzz) - Dom"), restando 6 abas, incluindo Canais Pagos("Canais Pagos(OCP) - Seg a Sex", "Canais Pagos(OCP) - Sab", "Canais Pagos(OCP) - Dom") e NI conteúdo, serviços de streaming que não são canais ou emissoras da televisão("NI Conteúdo - Seg a Sex", "NI Conteúdo - Sab", "NI Conteúdo - Dom"). Enquanto nas colunas, estão inclusas: "Dia"(XX/XX/XXXX), "Hora de Início", "Emissora", "Dia da Semana", várias delas relacionam o "Rat"(valor de audiência), "Shr%"(Share), "Rch%"(Reach), "Fid%"(Fidelidade), usando de base perfis de público(AB, C1, C2, DE, Masculino e Feminino) e diversos intervalos de faixa etária para avaliar esses 4 parâmetros.

O tamanho do arquivo que contém todos esses dados é de 437.754 KB.

Nome do atributo	Tipo do atributo	Descrição
Emissora	Texto	Empresa produtora e transmissora dos conteúdos daquela aba da tabela
Hora Início	Tempo	Horário que inicia a medição de audiência em uma determinada data
Data	Data	Data referência daquela medição de uma determinada emissora
Rat%(Rating)	Real	Valor da audiência dos programas
Shr%(Share)	Real	
Rch%(Reach)	Real	É o RAT e divide pelo TLE

Nome do atributo	Tipo do atributo	Descrição
Fid%(Fidelidade)	Real	É a fidelidade do público em relação a emissora
Faixa-Etária	Texto	Intervalo de idade do público telespectadores
Masculino	Texto	Intervalo que contém strings com a características
Feminino	Texto	Intervalo que contém strings com a características
AB	Texto	Intervalo da classe social A - B do público, associado a Rating, Share, Reach e Fidelidade
C1	Texto	Intervalo da classe social C do público associado a Rating, Share, Reach e Fidelidade
C2	Texto	Intervalo da classe social C2 do público associado a Rating, Share, Reach e Fidelidade
DE	Texto	Intervalo da classe social D - E do público associado a Rating, Share, Reach e Fidelidade
4-11 anos	Real	Intervalo de idade do público, associado aos atributos Rating, Share, Reach e Fidelidade
12-17 anos	Real	Intervalo de idade do público, associado aos atributos Rating, Share, Reach e Fidelidade
18-24 anos	Real	Intervalo de idade do público, associado aos atributos Rating, Share, Reach e Fidelidade
25-34 anos	Real	Intervalo de idade do público, associado aos atributos Rating, Share, Reach e Fidelidade
35-49 anos	Real	Intervalo de idade do público, associado aos atributos Rating, Share, Reach e Fidelidade
50-59 anos	Real	Intervalo de idade do público, associado aos atributos Rating, Share, Reach e Fidelidade
69+ anos	Real	Intervalo de idade do público, associado aos atributos Rating, Share, Reach e Fidelidade

Tabela 1 - Interpretação da descrição dos dados a serem utilizados

(Fonte: Planilha disponibilizada pelo parceiro de mercado).

a. **Se houver mais de um conjunto de dados, descrição de como serão agregados/mesclados.**

A organização dos dados foi realizada individualmente para as tabelas da Emissora Principal e Emissora Concorrente B, nas quais foram mescladas informações de todos os dias e finais de semana em uma única coluna, ordenados por data, horário de início e os respectivos programas e categorias da faixa horária disposta. Após isso, foram agregados os dados da Grade Diária em cada uma das tabelas. Ademais, foi-se adicionando uma coluna com o nome do mês e dia do mês para diferenciação na segmentação dos dados.

b. **Descrição dos riscos e contingências relacionados a esses dados (qualidade, cobertura/diversidade e acesso).**

- **Qualidade**

Para considerar dados que possuem qualidade, é iniciado a partir de três pilares: a integridade, a acuracidade e a completude. A integridade indica a segurança dos dados contidos na fonte. A acuracidade indica quanto os dados da fonte representam a realidade. A completude indica quanto de todos os dados necessários para atender a demanda do negócio está presente na fonte. Partindo desse pressuposto, o fato de não haver dados sobre identidade de gênero, raça, etnia e orientação, fazem com que sejam incompletos. Dados de qualidade eliminam problemas relativos ao negócio da organização como perda de receita, altos custos de produção, incapacidade de manter seus clientes fiéis, perda de mercado, dentre outros.

Os dados disponibilizados são limitados e não abrangem as informações citadas anteriormente, o que dificulta a obtenção de uma visão panorâmica, tal como extrair mais dados de pessoas que possuem os aparelhos em seus lares, já que os dados têm uma função fundamental na implementação de programas voltados para diversidade e inclusão como programações mais específicas que abrangem uma maior parcela da população, selecionando conteúdo para nichos específicos que interessam ao público.

Além disso, a pesquisa é feita por amostragem e não envolve toda a população da região e, por não ter muitos aparelhos instalados, os dados muitas vezes não são tão fiéis à realidade.

- **Diversidade**

Os dados abrangem pessoas do sexo feminino e masculino divididas em grupo nas seguintes faixa-etária:

➔ 60+ anos
➔ 50 a 59 anos
➔ 35 a 49 anos
➔ 25 a 34 anos
➔ 18 a 24 anos
➔ 12 a 17 anos

➔ 4 a 11 anos

- **A exibição da programação está dividida em:**

➔ Data;
➔ Hora de início;
➔ Emissora;
➔ Dia da semana;
➔ Quantidade de domicílios que assistem em rat% e share%;
➔ Classe do espectador dividida em AB, C1, C2

- **Acesso**

O aparelho chamado PeopleMeter, a medida que identifica o canal que o espectador está acompanhando, envia informações diariamente para a central através do sinal de radiofrequência. Os dados referente a audiência são disponibilizados através do Instituto Brasileiro de Opinião Pública e Estatística, o IBOPE, que posteriormente são tabulados e repassados para as emissoras de TV que possuem acesso aos resultados.

c. Se aplicável: descrição de como será selecionado o subconjunto para análises iniciais (quando o tamanho do conjunto de dados impossibilita a utilização do conjunto completo em todas as etapas da definição do modelo a ser usado).

A análise inicial será limitada a emissora da empresa parceira e uma concorrente direta. O subconjunto selecionado será a média dos atributos da tabela por sessões de uma hora para cada dia de uma única semana do mês durante o período de 24 meses (2 anos).

d. Se houver: descrição das restrições de segurança.

A Rede Gazeta e a Kantar Ibope possuem um contrato de confidencialidade que veta a divulgação do nome das emissoras. Portanto, usa-se nomes fantasias (exemplos: Concorrente 1, Concorrente 2 e etc).

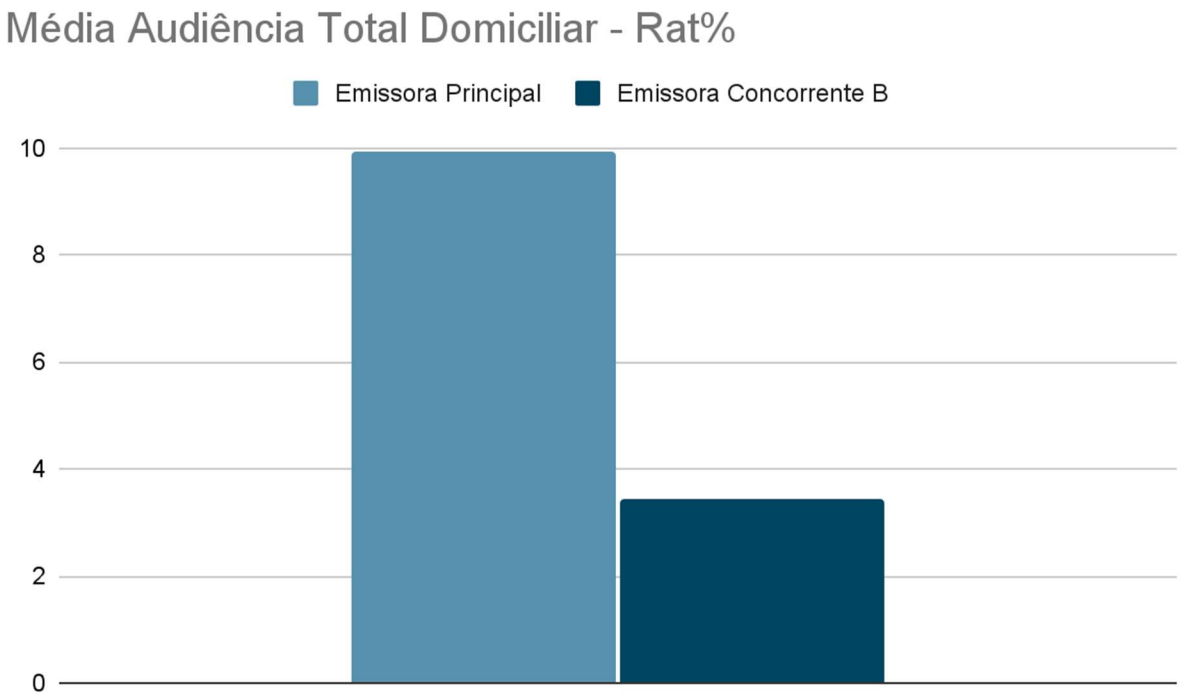
2. Descrição estatística básica dos dados, principalmente dos atributos de interesse, com inclusão de visualizações gráficas e como essas análises embasam suas hipóteses.

Análise Descritiva:

Dados: Audiência por Total Domiciliar(Rat%) Emissora Principal x Emissora concorrente B

Métrica	Emissora Principal	Emissora Concorrente B

Média Audiência Total Domiciliar %	9.92	3.45
Moda Audiência Total Domiciliar %	3.72	0.0
Mediana Audiência Total Domiciliar %	8.41	2.87
Máximo Audiência Total Domiciliar %	45.34	37.7
Mínimo Audiência Total Domiciliar %	0.0	0.0



Comparando os dados anteriores, é notória a percepção de 6.47 pontos de diferença, na audiência média aferida, entre a Emissora Principal e a Emissora Concorrente B. Isso demonstra, de forma nítida, a discrepância entre o índice de audiência no que se refere aos telespectadores, das duas emissoras comparadas, presente até nos mais altos picos de medida.

Dados: Audiência por Gênero (Rat%) -Emissora Principal x Emissora concorrente B

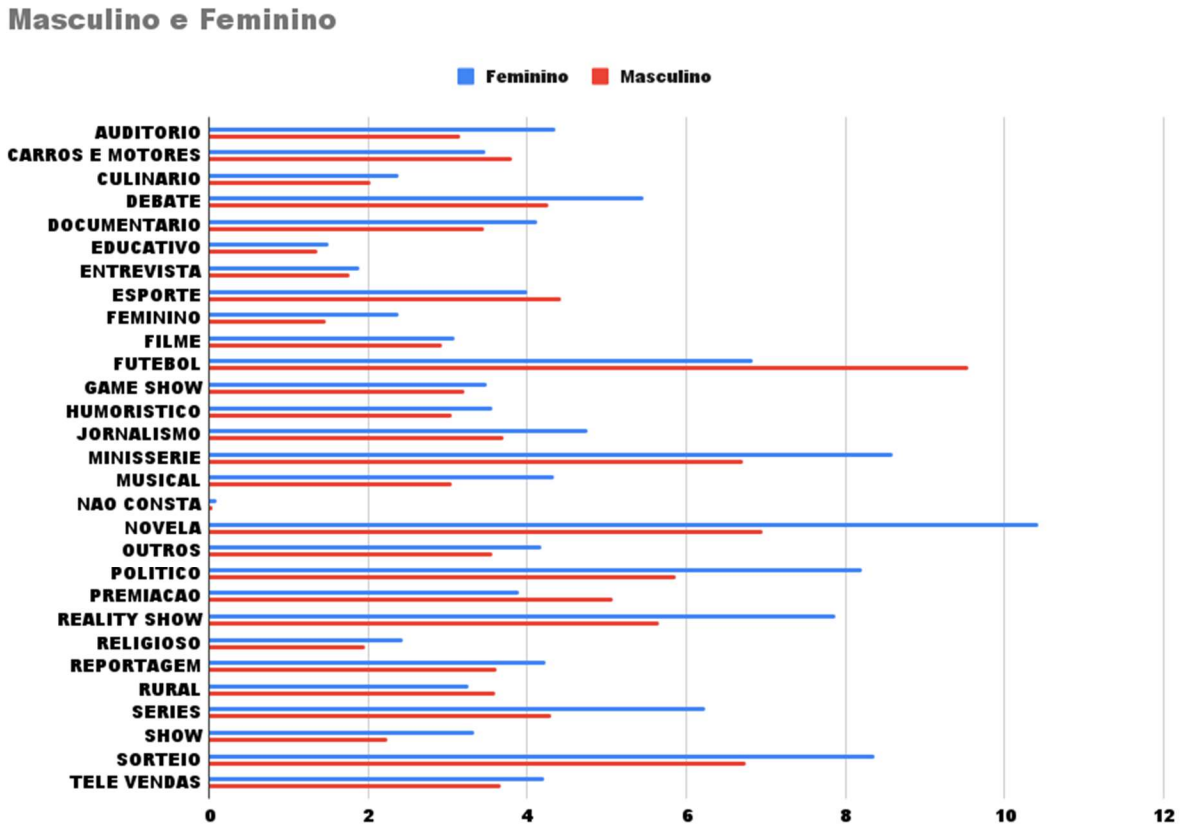
Métrica	Emissora Principal	Emissora Concorrente B
Média de Audiência por Gênero Feminino (Rat%)	5.14	1.84
Média de Audiência por Gênero Masculino (Rat%)	4.03	1.37

Moda Audiência por Gênero Feminino (Rat%)	0.0	0.0
Moda Audiência por Gênero Masculino (Rat%)	0.0	0.0
Mediana Audiência por Gênero Feminino (Rat%)	4.08	1.54
Mediana Audiência por Gênero Masculino (Rat%)	3.41	1.14
Máximo Audiência por Gênero Feminino (Rat%)	24.40	16.59
Máximo Audiência por Gênero Masculino (Rat%)	24.09	20.68
Mínimo Audiência por Gênero Feminino (Rat%)	0.0	0.0
Mínimo Audiência por Gênero Masculino (Rat%)	0.0	0.0

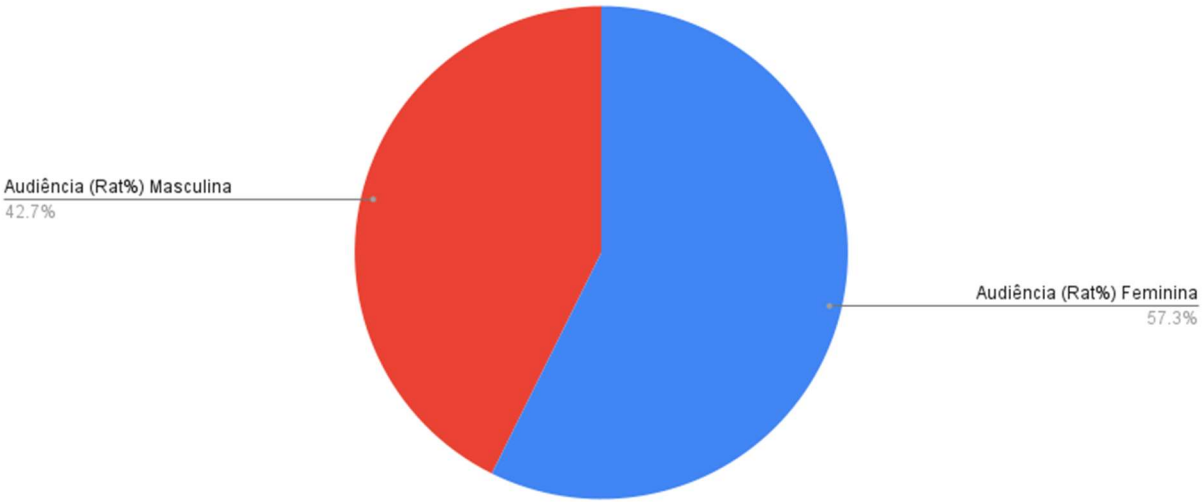
A partir do cálculo das médias das audiências masculinas e femininas, chegou-se à conclusão de que na Emissora Principal, é predominante a presença do público feminino, compondo aproximadamente 56.1% do público total. Para este cálculo, foram comparadas as médias dos valores Rat% de cada gênero nos últimos dois anos. Fazendo uma comparação direta entre elas, chegou-se neste gráfico abaixo.

Porcentagem do Público Feminino e Masculino - Emissora Principal:

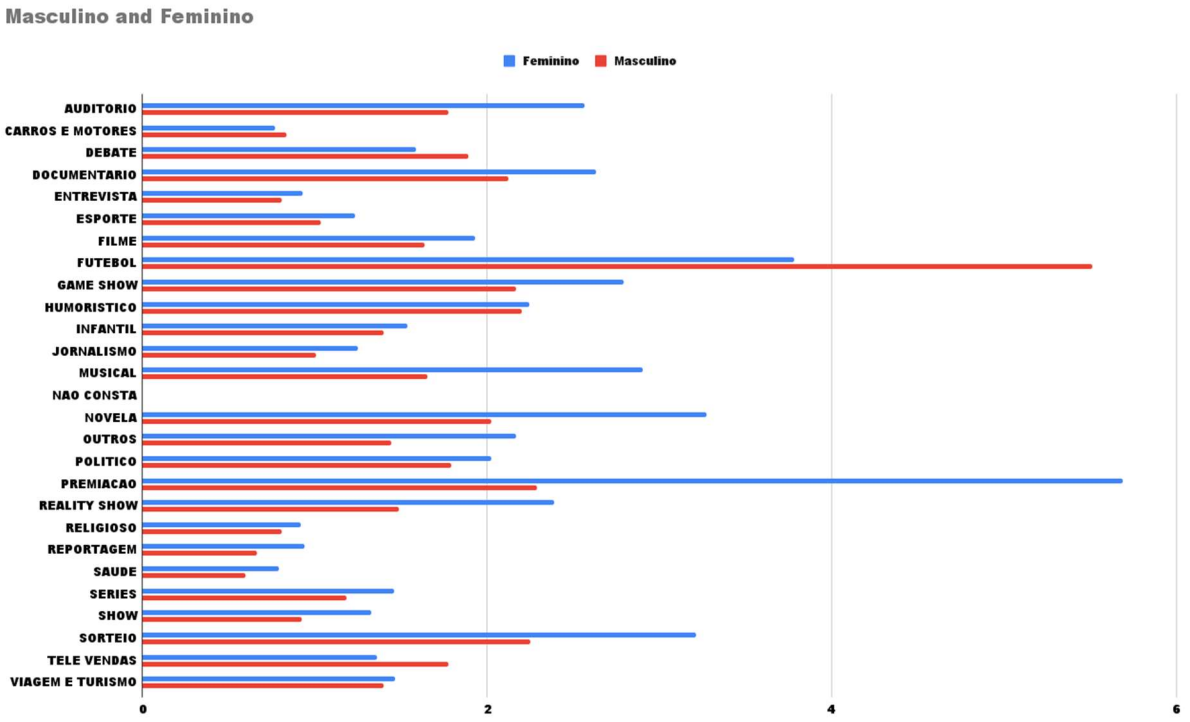
Público Feminino e Masculino em Relação às Categoria - Emissora Principal



Porcentagem do Público Feminino e Masculino - Emissora Concorrente B



Público Feminino e Masculino em Relação às Categoria - Emissora Concorrente B

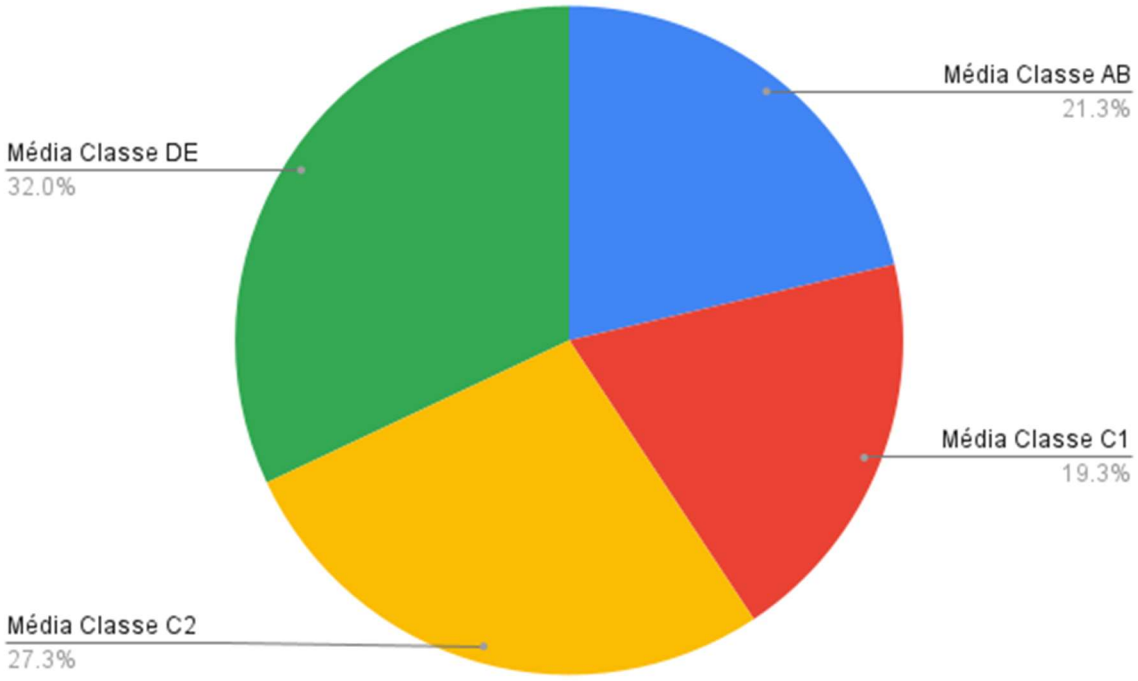


Dados: Audiência por Classe Social (Rat%) Emissora Principal x Emissora concorrente B

Métrica	Emissora Principal	Emissora Concorrente
Média de Audiência por Classe Social AB (Rat%)	4.03	0.89
Média de Audiência por Classe Social C1 (Rat%)	3.65	1.20
Média de Audiência por Classe Social C2 (Rat%)	5.16	1.34
Média de Audiência por Classe Social DE (Rat%)	6.04	3.87
Moda Audiência Classe Social AB (Rat%)	0.0	0.0
Moda Audiência Classe Social C1 (Rat%)	0.0	0.0

Moda Audiência Classe Social C2 (Rat%)	0.0	0.0
Moda Audiência Classe Social DE (Rat%)	0.0	0.0
Mediana Audiência Classe Social AB (Rat%)	3.51	0.58
Mediana Audiência Classe Social C1 (Rat%)	2.96	0.76
Mediana Audiência Classe Social C2 (Rat%)	3.95	0.85
Mediana Audiência Classe Social DE (Rat%)	4.70	3.43
Máximo Audiência Classe Social AB (Rat%)	22.28	16.31
Máximo Audiência Classe Social C1 (Rat%)	22.40	23.78
Máximo Audiência Classe Social C2 (Rat%)	29.83	17.44
Máximo Audiência Classe Social DE (Rat%)	31.44	25.96
Mínimo Audiência Classe Social AB (Rat%)	0.0	0.0
Mínimo Audiência Classe Social C1 (Rat%)	0.0	0.0
Mínimo Audiência Classe Social C2 (Rat%)	0.0	0.0
Mínimo Audiência Classe Social DE (Rat%)	0.0	0.0

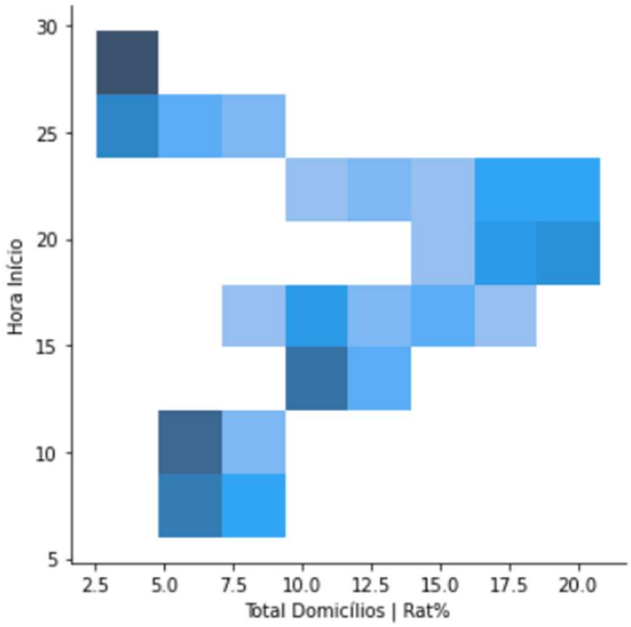
Público Dividido em Relação às Classe Sociais - Emissora Principal:



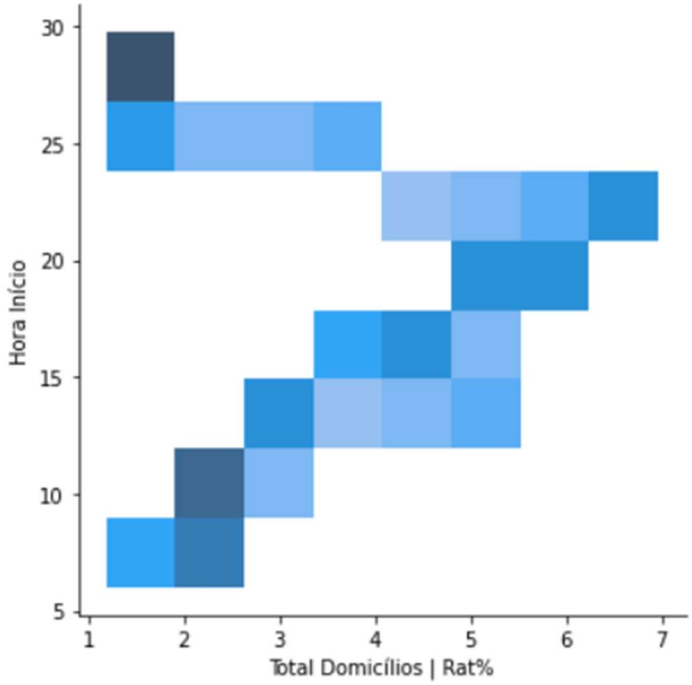
Referente ao seccionamento de classes, e sua distribuição na audiência total da Emissora Principal, verifica-se que a classe D e E possuem maior participação (32.0%), seguida pela classe C2 (27.3%), classes A e B (21.3%), em último lugar, a classe C1 (19.3%). Percebe-se que, em geral, há uma proximidade muito grande entre as médias das classes A e B, e a da classe C1, assim como a das classes D e E e as classes C2. Isso, de certa forma, reflete a realidade Brasileira atual, onde, segundo um levantamento feito pela Consultoria Tendências, é revelado que as classes D e E são mais da metade das casas no Brasil.

Dados: Concentração de Taxas de Audiência por Hora de Início (Rat%) Mapa de Calor

Mapa de Calor - Emissora Principal



Mapa de Calor - Emissora Concorrente B



Comparando a audiência total, por hora de início, percebe-se uma grande concentração de altas taxas de audiência Total Domiciliar, no período próximo entre 18h e 20h. Ao comparar a grade de eventos da Emissora Principal, nota-se uma grande presença de episódios de novela, intermitentes nesta faixa. Logo em seguida, das 20h até as 22h, há uma leve queda nesta medida, mas permanece a alta concentração de audiência. Extraindo a grade desta última faixa, encontra-se por quase todo o momento, o programa Jornal Nacional. Além disso, é também perceptível que, no mesmo período de alta, a emissora concorrente B apresenta uma queda considerável em sua concentração.

Dados: Audiência por Faixa Etária(Rat%) Emissora Principal x Emissora concorrente B

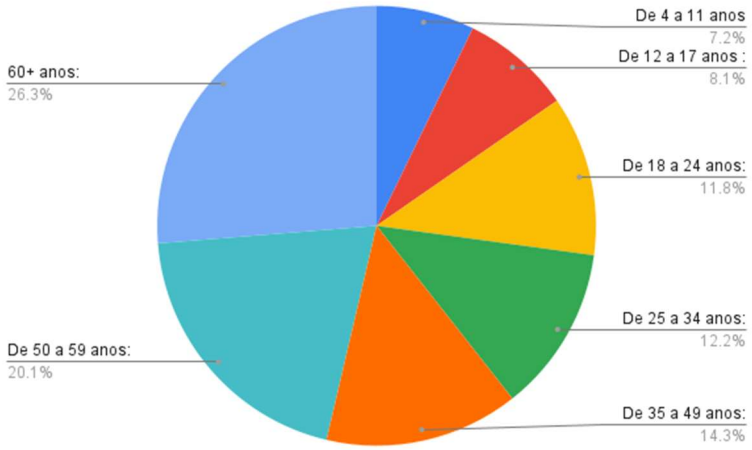
Métrica	Emissora Principal	Emissora Concorrente
Média de Audiência por Faixa Etária 4 - 11 anos (Rat%)	2.2	1.68
Média de Audiência por Faixa Etária 12 - 17 anos (Rat%)	2.51	1.32
Média de Audiência por Faixa Etária 18 - 24 anos (Rat%)	3.63	1.07
Média de Audiência por Faixa Etária 25 - 34 anos (Rat%)	3.77	1.31
Média de Audiência por Faixa Etária 35 - 49 anos (Rat%)	4.40	1.31
Média de Audiência por Faixa Etária 50 - 59 (Rat%)	6.18	1.94
Média de Audiência por Faixa Etária 60+ anos (Rat%)	8.10	2.60
Moda de Audiência por Faixa Etária 4 - 11 anos (Rat%)	0.0	0.0
Moda de Audiência por Faixa Etária 12 - 17 anos (Rat%)	0.0	0.0
Moda de Audiência por Faixa Etária 18 - 24 anos (Rat%)	0.0	0.0
Moda de Audiência por Faixa Etária 25 - 34 anos (Rat%)	0.0	0.0
Moda de Audiência por Faixa Etária 35 - 49 anos (Rat%)	0.0	0.0
Moda de Audiência por Faixa Etária 50 - 59 anos (Rat%)	0.0	0.0
Moda de Audiência por Faixa Etária 60+ anos (Rat%)	0.0	0.0
Mediana Audiência por Faixa Etária 4 - 11 anos (Rat%)	1.47	1.04

Mediana de Audiência por Faixa Etária 12 - 17 anos (Rat%)	1.57	0.22
Mediana de Audiência por Faixa Etária 18 - 24 anos (Rat%)	2.61	0.00
Mediana de Audiência por Faixa Etária 25 - 34 anos (Rat%)	2.89	0.98
Mediana de Audiência por Faixa Etária 35 - 49 anos (Rat%)	3.78	0.94
Mediana de Audiência por Faixa Etária 50 - 59 anos (Rat%)	5.01	1.49
Mediana de Audiência por Faixa Etária 60+ anos (Rat%)	6.68	1.81
Máximo de Audiência por Faixa Etária 4 - 11 anos (Rat%)	20.25	16.82
Máximo de Audiência por Faixa Etária 12 - 17 anos (Rat%)	22.72	18.57
Máximo de Audiência por Faixa Etária 18 - 24 anos (Rat%)	28.07	16.68
Máximo de Audiência por Faixa Etária 25 - 34 anos (Rat%)	30.15	28.73
Máximo de Audiência por Faixa Etária 35 - 49 anos (Rat%)	21.90	21.07
Máximo de Audiência por Faixa Etária 50 - 59 anos (Rat%)	32.51	19.29
Máximo de Audiência por Faixa Etária 60+ anos (Rat%)	37.64	24.53
Mínimo de Audiência por Faixa Etária 4 - 11 anos (Rat%)	0.0	0.0
Mínimo de Audiência por Faixa Etária 12 - 17 anos (Rat%)	0.0	0.0
Mínimo de Audiência por Faixa Etária 18 - 24 anos (Rat%)	0.0	0.0
Mínimo de Audiência por Faixa Etária 35 - 49 anos (Rat%)	0.0	0.0

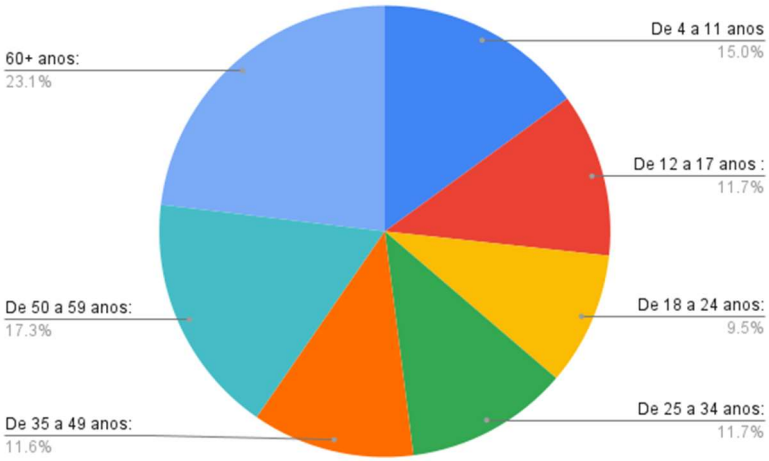
Mínimo de Audiência por Faixa Etária 50 - 59 anos (Rat%)	0.0	0.0
Mínimo de Audiência por Faixa Etária 60+ anos (Rat%)	0.0	0.0

A partir da análise das médias de audiência por faixa etária, infere-se que a maior parcela da audiência da rede, 25.7%, se deve à pessoas com 60 ou mais anos. Em segundo lugar, posicionam-se pessoas entre 50 a 59 anos, com 20.3% da parcela. Seguido por pessoas com 35 a 49 anos(14.3%), 25 a 34 anos (12.3%), 18 a 24 anos (12.0%), 12 a 17 anos (8.3%) e 4 a 11 anos (7.2%). Com isso, conclui-se que a audiência predominante é a da população idosa, seguida pela meia-idade.

Público Dividido em Relação às Faixa Etária - Emissora Principal



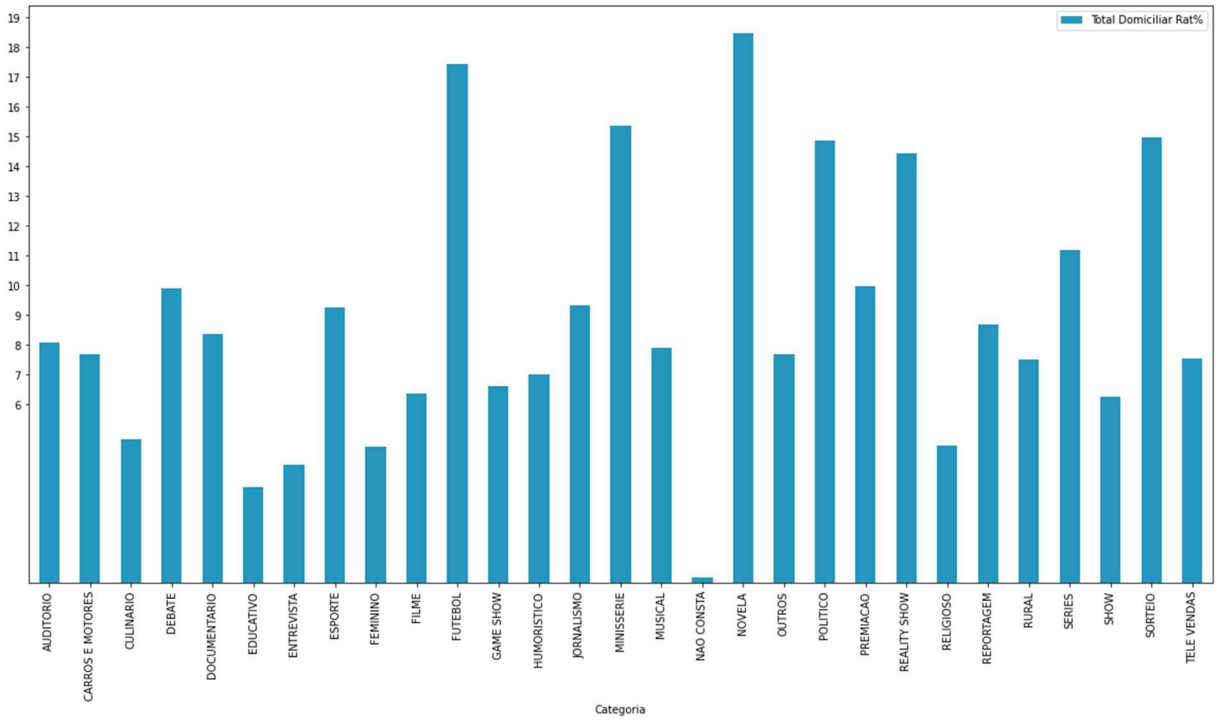
Público Dividido em Relação às Classe Sociais - Emissora Concorrente B



Dados: Audiência por Programação de Conteúdos Categorizados (Rat%) Emissora Principal x Emissora Concorrente B

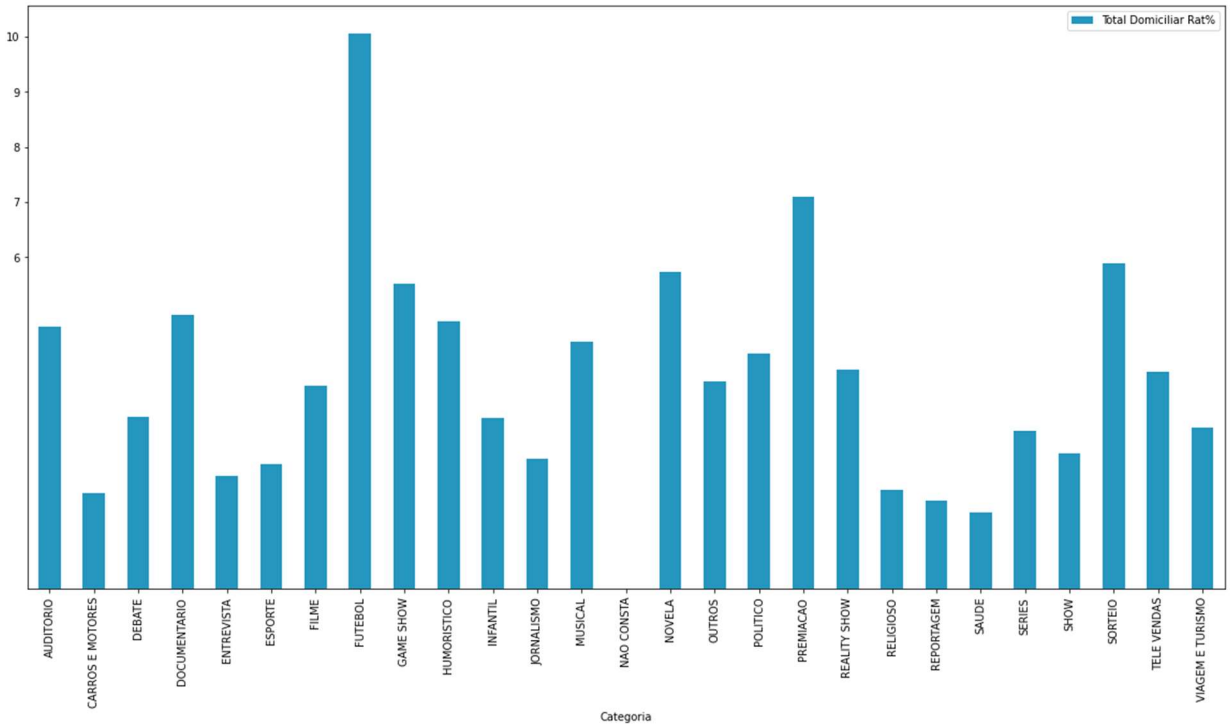
Categoria	Pontos Emissora Principal	Pontos Emissora Concorrente B
Novela	18.49	5.73
Futebol	17.44	10.05
Premiação	9.99	7.08
Sorteio	14.99	5.89
Auditório	8.09	4.75
Tele Vendas	7.53	3.93
Game Show	6.61	5.52
Político	14.86	4.25
Carro e Motores	5.89	1.74

Dados de Audiência em Relação a Conteúdos Categorizados - Emissora Principal



Nesse sentido, há uma relação direta entre as categorias ditas na tabela e a aproximação que o gráfico trouxe. É perceptível, que programas de entretenimento como “FUTEBOL”e “NOVELA” lideram os dados, deixando o “JORNALISMO” em terceiro lugar.

Dados de Audiência em Relação a Conteúdos Categorizados - Emissora Concorrente B



Levando em consideração também, a audiência por categoria, da emissora concorrente B, é visível a alta demanda pela seção “FUTEBOL”, o qual é predominante sobre todas as outras seções. Diferentemente da emissora principal, o conteúdo “NOVELA” se encontra em terceira posição, perdendo lugar para a categoria “PREMIAÇÃO”.

3. **Descrição da predição desejada (“target”), identificando sua natureza (binária, contínua, etc.)**

A predição utilizará target de dados quantitativos de audiência fornecidos pela coluna de valor de audiência (Rat%). Os dados são compostos por números decimais e o modelo é de natureza contínua, buscando-se estimar o valor mais provável do Rat%. É esperado que ao final da construção sistema preditivo, por meio da aplicabilidade do input destes conjuntos de dados de entrada, a resposta predita apresentada pelo algoritmo seja a mais provável de ocorrer para a data e horário selecionada pelo usuário.

4.3. Preparação dos Dados

4.3.1 Anonimização dos dados

Colab de referência:
https://drive.google.com/file/d/1nrO57J8Xk09h0iDu6xkwPqAh0sSkm5mz/view?usp=sharing

Devido a limitações contratuais do parceiro e com a Kantar Ibope, os nomes das emissoras e quaisquer menções diretas, foram substituídas por referências anonimizadas.

Para isso, o seguinte processo foi efetuado:

Substituição dos nomes de emissoras por referências anonimizadas
Célula de Referência: 1.2 Renomeação das colunas sensíveis

Descrição: As colunas da planilha Grade Diária foram alteradas por correspondentes respectivos, anonimizados, transformando-se em “Emissora A”, “Emissora B” e “Emissora C”.

Substituição da programação por referências anonimizadas
Célula de Referência: 1.3 Anonimização dos nomes dos programas

Descrição: Devido à programações específicas de emissoras, foram renomeados os programas por um dicionário respectivo, alterando elementos como “JORNAL ABC” para “programa_n”, sendo “n”, uma representação no dicionário de programas. Isto foi feito utilizando o método de regex, para encontrar e substituir o nome dos programas, sem alterar a estrutura da tabela original. Além disso, a coluna “Praça” foi substituída da mesma forma, por conter dados confidenciais de identificação das emissoras.

Remoção da coluna “Emissora”
Célula de Referência: 2.1 Remoção da coluna Emissora das Tabelas

Descrição: Na planilha de audiência da emissora principal, os arquivos foram renomeados e a coluna “Emissora” foi removida.

4.3.2 Manipulação de dados e registros

Transformação da coluna “Faixa Horária” em “Hora Início"
Célula de Referência: 1.2 Transformação da coluna “Faixa Horária” em “Hora Início”.

Descrição: Divisão do horário por " - "(por meio de um split, que basicamente transforma uma string em lista, permitindo assim a sua divisão em duas ou mais partes) na feature "Faixa Horária". Foi criado a "Hora Início" como sendo “coluna filha” da coluna "Faixa Horária", atribuindo a ela a padronização de 00:00:00 (hora:minuto:segundo), por meio da concatenação do horário inicial (primeiro elemento extraído do split) + ':00'.

Objetivo: Ter uma padronização uniforme dos dados horários estabelecidos.

Conversão de tipos da coluna “Data”
Célula de Referência: 1.3 Conversão de tipos da coluna “Data”

Descrição: Conversão da coluna “Data” do tipo string para o tipo datetime.
Objetivo: Possibilitar a ordenação correta da coluna “Data” por meio do sort_values (que seria usado na célula 2.4) para considerar a ordem dos valores de datetime.

Derivação dos atributos “Programa” e “Categoria”
Célula de Referência: 1.4 Derivação dos atributos “Programa” e “Categoria”.

Descrição: Divisão da coluna das emissoras para as colunas "Programa" e "Categoria", aplicando isso à emissora “A".
Objetivo: Selecionar as features, para facilitar o filtro das informações que pretende-se obter de acordo com cada emissora desejada.

Concatenação das tabelas dos Dias da Semana - Emissora A
Célula de Referência: 2.2 Concatenação das tabelas dos Dias da Semana - Emissora A.

Descrição: Concatenação (usando o método concat) das tabelas A - Seg a Sex.csv, A - Sab.csv e A - Dom.csv
Objetivo: Concentrar a audiência de todos os dias da semana em uma mesma tabela, possibilitando a comparação assertiva com a organização dos dias da semana na tabela Grade Diária.

Conversão de colunas “Data” para “Datetime”
Célula de Referência: 2.3 Conversão de tabelas de “Data” para “Datetime” **.**

Descrição: Conversão da coluna "Data'" das três tabelas concatenadas, do tipo string para o tipo “datetime”
Objetivo: Fornecer uma padronização para manipulações futuras, como a ordenação correta da coluna “Data” por meio do sort_values, para considerar a ordem dos valores de datetime.

Sort da tabela agregada “emissora_a_geral” pelas colunas “Data” e “Hora Início”
Célula de Referência: 2.4 Ordenamento da tabela agregada “emissora_a_geral” pelas colunas “Data” e “Hora Início”.

Descrição: Ordenação das colunas com base nas colunas "Data" e "Hora Início", por meio do método “sort_values”, que permite a ordenação, já que as colunas foram convertidas de string para datetime.

Objetivo: Ordenar dados da tabela que foram concatenados, inicialmente pela ordem das datas e levando em conta a coluna “Hora Início”.

Agregação das duas tabelas manipuladas: “Grade Diária” e “emissora_a_geral”
Célula de Referência: 3.1 Agregação (Merge) das duas tabelas Manipuladas: "Grade Diária" e "Emissora A".

Descrição: Merge das duas tabelas manipuladas, tanto a de "dataset_grade_diaria" (tabela inicial), quanto à "_geral" (concatenação das três tabelas) com base nas features "Data" e "Hora Início". Também usando o método drop, para a remoção de colunas não utilizadas.
Objetivo: Unificar as duas tabelas para conseguir unir a categoria (que está contida na tabela de grade diária) com a audiência (que está contida na tabela da emissora “A” geral).

Limpeza de dados
Célula de Referência: 3.2 Limpeza de dados.

Descrição: Remoção de campos vazios, nulos, ou categorizados como “Não Consta”.
Objetivo: Deixar os dados mais enxutos, tirando campos sem utilidade para o modelo.

Derivação de novos atributos: Mês e Dia a partir da coluna “Data”
Célula de Referência: 3.3 Derivação de novos atributos: Mês e Dia a partir da coluna “Data”.

Descrição: Separação da coluna "Data" para novas colunas "Mês" e "Dia".
Objetivo: Poder associar essas colunas com o mês e dia de um programa que será lançado, permitindo um modelo com mais clareza.

Conversão das Colunas para o tipo “Int”
Célula de Referência: 3.4 Conversão das Colunas para o tipo “Int”.

Descrição: Transformação da Hora Início, em intervalos de 15 minutos, mudando o número de acordo com o intervalo. Transformação da nomenclatura dos dias da semana em números para entendimento do programa.
Objetivo: Poder utilizar as colunas, a partir dos dicionários data e dia da semana, como novos inputs no futuro.

Utilização do Método One-Hot-Encoding para tratar os dados categorizados (Categoria)
Célula de Referência: 3.5 Utilização do Método One-Hot-Encoding para tratar os dados categorizados (Categoria).

Descrição: Cada categoria é adaptada para uma coluna, assim o sistema entende que se naquele instante (dentro do intervalo de 15 minutos) determinada categoria estiver passando, ela será contabilizada com 1 (verdadeiro) e se não estiver, será 0 (falsa).
Objetivo: Associar cada categoria com um número, para o programa interpretar o que está acontecendo.

Agregação da tabela ”Feriados” com a tabela “Emissora A”
Célula de Referência: 4. Agregação da tabela “Feriados” com a tabela “Emissora A”.

Descrição:
Data: Transformação de “string” para “datetime”.
Coluna feriado: Foi transformado os valores da coluna feriado em números booleanos.
Objetivo: Trazer os feriados nacionais para incluir como coluna no modelo, tendo assim outra variável, buscando uma melhor predição.

Separação das colunas em features e labels
Célula de Referência: 5.1 Seleção das Features e Labels.

Descrição: Foram criadas duas variáveis, x e y , que armazenam, respectivamente, as features selecionadas para utilização do modelo de regressão, tais como: “Hora Início”, “Dia da Semana”, “Mês”, “Dia”, “Feriado”, Categorias, e outras, além de labels para a tabela final, como por exemplo: “ Rat% Total Domicílios”, Rat% por classe social e Rat% por gêneros.
Objetivo: Ter um núcleo com todas as informações selecionadas, para obter as informações necessárias para a predição de performance de um novo programa.

4.3.3 Agregação de Registros e derivação de novos atributos

O processo de agregação de registro e derivação de novos atributos foi realizado por etapas, por meio da mesclagem de planilhas selecionadas para o modelo preditivo. As planilhas utilizadas foram: “Planilha Grade Diária”, “Planilha Emissora A Geral”, “Planilha Programação” e "Planilha Feriados Nacionais”.

Merge das planilhas

O processo de mesclagem das planilhas foi feito por etapas, começando pela junção das planilhas da Grade Diária e Emissora Geral A, seguindo para a Planilha Programação, e por último a Planilha feriados nacionais. A técnica utilizada foi “left merge”, função pd.merge, na qual se mantém todos os dados da planilha à esquerda, e adiciona-se a nova planilha à direita, transformando suas linhas inexistentes em valores NaN.

Exemplo de código utilizado para mesclagem das tabelas:

```
tabela_programacao = pd.merge(emissora_A_geral, dataset_gradediaria, how="left", left_on=['Data', 'Hora Início',], right_on =['Data', 'Hora Início'])
```

Concatenação

Foi-se aplicado o método concat() para juntar dados das planilhas de seg-sex, sábado e domingo. A concatenação agrupou as três planilhas verticalmente, por coluna, de forma não ordenada.

Código utilizado para concatenação dos dataframes:

```
tv_emisssora_A = pd.concat([tv_emissora_A, tv_emissora_A_sab, tv_emissora_A_dom])
```

Derivação de novos atributos

Para o processo de derivação de novos atributos de outras colunas, foi-se aplicado a função dt.strftime(%), a qual cria uma nova coluna a partir de um valor que está no formato datetime. Também foi utilizado a função .split , para separar strings e posteriormente converter para datetime, visando a extração eficiente e formatação dos dados para saída e manipulação.

Exemplo do código utilizado para derivação de novos atributos:

```
tabela_programacao['Mês'] = tabela_programacao['Data'].dt.strftime('%m')
```

One-hot encoding

O método foi aplicado para transformar os valores dos atributos em formato de números binários, aplicando a lógica booleana. A função do pandas utilizada foi pd.get_dummies.

Exemplo do código utilizado no método:

```
Tabela_convertido = pd.get_dummies(tabela_convertido,columns=['Categoria'], drop_first=True)
```

Dicionários

Os dicionários foram criados para agregar intervalos de string em um único número, na coluna de Horas e Dia da semana.

Horas: Os intervalos de 15 minutos foram separados em um único número, para identificar aquele horário. Períodos de Xh até Xh10m foram identificados como X; Xh15m até Xh25m foram identificados como X,25; Xh30m até Xh40m foram identificados como X,5; Xh45 até Xh55 foram identificados como X,75.

Dia da semana: Foi numerado de 0 até 6, começando por domingo.

Exemplo do código utilizado:

```
dicionario_dia_semana = {"Domingo":0,"Segunda":1,"Terça":2,"Quarta":3,"Quinta":4,"Sexta":5,"Sábado":6}
```

4.3.4 Remoção e substituição de valores ausentes, em branco, ou desconsiderados

Ao vislumbrar o dataframe, não foram encontradas quaisquer linhas ou colunas com valores nulos ou ausentes, porém, durante a análise, foi verificada a categoria “Não Consta”, considerada como valor ausente, pois não havia nenhuma informação referente a programação. Ademais, como descrito na tabela, foi concluído que essa categoria poderia ser definida como um momento de transmissão ausente, indefinido ou que há apenas a logo da emissora e, por apresentar pouquíssimas quantidades de valores ao comparar com o total da tabela, esse tipo categórico foi considerado irrelevante.

Desse modo, para conhecer e visualizar todos os valores da coluna categoria, foi utilizada a biblioteca Pandas e funções para converter os valores em uma lista, para assim, confirmar se os valores da coluna buscada estavam realmente presentes. Para prosseguir com o processo de remoção, foi efetuado o “drop” e, a partir disso, removida cada linha especificada que obtém o valor “Não Consta”, sem obter uma cópia, no final, filtrando os dados resultantes para ter uma parte da visualização da tabela. Portanto, as funções necessárias para visualizar os valores, assim como o “drop” para realização da remoção das linhas, estão disponíveis nas células textuais do Google Colab.

4.3.5 Identificação das features selecionadas

Como características padronizadas de um novo modelo, de um programa, em que abordam valores de entrada, foram selecionados as seguintes features que se encaixam com esses valores:

Hora Início

Utilizado para separar o intervalo de tempo da predição do modelo preditivo, a qual será determinada em intervalos de 15 em 15 minutos, ao invés de a cada 5 minutos, pois assim, há um maior aproveitamento do tempo e mais dados são encapsulados devido a duração estendida.

Dia da Semana, Mês e Dia

As três features, em conjunto, serão utilizadas para identificar os respectivos “dia da semana”, “mês” e “dia” para qual a audiência da predição será calculada. Estas foram formatadas e divididas em colunas diferentes, para possibilitar a leitura dos valores pelo sistema.

Feriado

Os valores das colunas foram transformados em números booleanos, para a leitura do sistema, e integrados com a tabela original. Essa integração foi considerada, pois ocorrem mudanças na grade da programação, e os feriados possuem influência positiva na audiência, apresentando-se relevantes.

Categoria

Anteriormente, as categorias estavam em formato string e em linhas. Foi executado processos de formatação e conversão das linhas categóricas, em colunas, assim como a remoção da primeira linha da tabela chamada "categoria", não mais necessária. Os valores que fazem parte dessa categorização, foram convertidos em números de identificação, para possibilitar a leitura pelo sistema. Por exemplo:

- Categoria 1: valor 1;
- Categoria 2: valor 2;
- Categoria 3: valor 3.

Desse modo, totalizou-se 46 colunas, utilizando os tipos de categorização específicas para cada programa na grade.

Labels

Para obter a audiência dos programas, é efetuada a medição pelo Rat%, e os labels são os resultados que devem ser devolvidos, servindo como um objeto extra a ser inserido. Não são features, mas são outputs/targets esperados para o modelo desenvolvido. Os labels utilizados são:

- 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 I Rat%.

4.4. Modelagem

RFE - Recursive Feature Elimination

5.3 Ordenação de Features por importância - Árvore de Decisão - RFE

Buscando entender a participação e peso de cada feature utilizada, foi aplicado um método recursivo de eliminação e ordenação de features, para as colunas-alvo "Rat". Baseando-se no modelo de árvore de decisão regressivo, definiu-se três cenários: O primeiro, contendo todas as features X especificadas, o segundo contendo somente as categorias de programas televisivos, e o terceiro contendo apenas Hora Início, Dia da Semana, Mês, Dia do Mês e Feriado.

Ranqueamento de Todas as Features

Feature	Ranqueamento
Hora Início	1
Categoria_NOVELA	2
Dia	3
Dia da Semana	4
Mês	5
Categoria_JORNALISMO	6
Categoria_FILME	7
Categoria_FUTEBOL	8
Categoria_REALITY SHOW	9
Feriado	10
Categoria_ESPORTE	11
Categoria_SERIES	12
Categoria_SHOW	13
Categoria_HUMORISTICO	14
Categoria_REPORTAGEM	15
Categoria_POLITICO	16
Categoria_MUSICAL	17
Categoria_ENTREVISTA	18
Categoria_FEMININO	19
Categoria_MINISSERIE	20

Categoria_RELIGIOSO	21
Categoria_GAME SHOW	22
Categoria_CARROS E MOTORES	23
Categoria_RURAL	24
Categoria_DOCUMENTARIO	25
Categoria_EDUCATIVO	26
Categoria_PREMIACAO	27
Categoria_CULINARIO	28
Categoria_OUTROS	29
Categoria_DEBATE	30
Categoria_TELE VENDAS	31
Categoria_SORTEIO	32

Ranqueamento das features (sem categoria)

Feature	Ranqueamento
Hora Início	1
Dia	2
Dia da Semana	3
Mês	4
Feriado	5

Ranqueamento das features (somente categoria)

Feature	Ranqueamento
Categoria_NOVELA	1
Categoria_FUTEBOL	2
Categoria_FILME	3
Categoria_REALITY SHOW	4
Categoria_SERIES	5
Categoria_JORNALISMO	6

Categoria_ESPORTE	7
Categoria_ENTREVISTA	8
Categoria_EDUCATIVO	9
Categoria_SHOW	10
Categoria_FEMININO	11
Categoria_RELIGIOSO	12
Categoria_POLITICO	13
Categoria_MINISSERIE	14
Categoria_REPORTAGEM	15
Categoria_HUMORISTICO	16
Categoria_CARROS E MOTORES	17
Categoria_CULINARIO	18
Categoria_GAME SHOW	19
Categoria_MUSICAL	20
Categoria_RURAL	21
Categoria_PREMIACAO	22
Categoria_DOCUMENTARIO	23
Categoria_SORTEIO	24
Categoria_DEBATE	25
Categoria_TELE VENDAS	26
Categoria_OUTROS	27

Modelo de Regressão linear

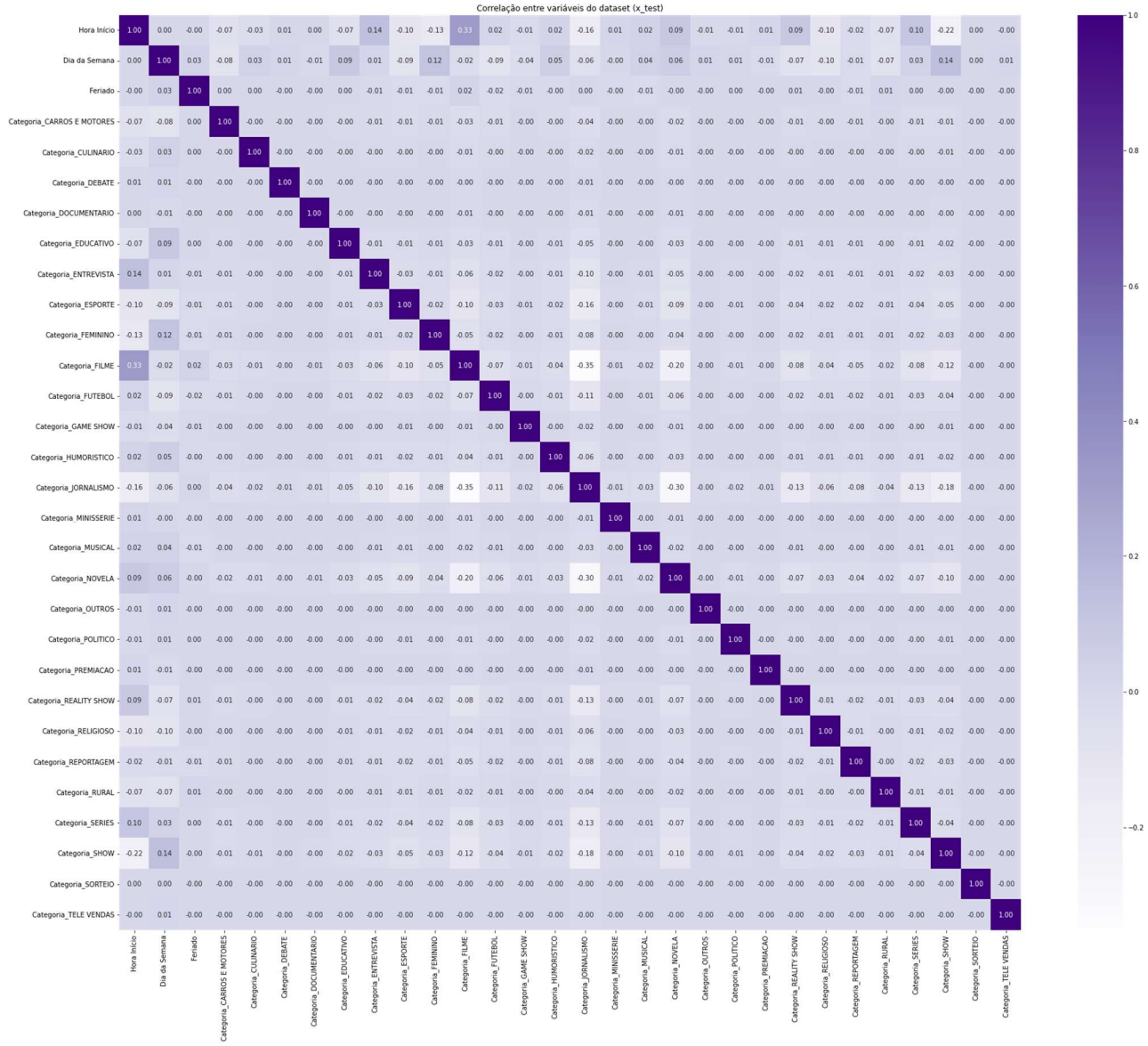
Seção de referência: 5.4.1 Regressão Linear

Nos testes preliminares, como parte dos tipos de modelos a serem escolhidos, foi treinado e testado o modelo de regressão linear para verificar o seu ajuste e acuracidade. Nesse caso, como esse tipo de modelo possui uma relação linear entre as variáveis, consequentemente, não há um ajuste suficiente destas. Desse modo, para comprovar se esse modelo poderia ser a melhor opção, são seguidos os passos abaixo:

1. Como esperado, foi dividido os valores em variáveis x e y e utilizado estas em treino e teste para posteriormente realizar a predição utilizando a biblioteca sklearn.
2. A estratégia de escolha para medir a precisão do teste e do treino foi o R-Squared. Sendo assim, a precisão do treino obtido foi de 0.4170, enquanto a do teste foi de 0.4160. Aqui, a precisão foi considerada como baixa, pois, o valor obtido é abaixo de 0.5 e está longe de 1.0. Esse resultado obtido foi apenas a partir do uso do "total de domicílios Rat%"
3. Para avaliar a precisão do modelo foi usado a biblioteca Statsmodels, aqui para o valor do R-Squared obtém-se um resultado de 0.002, ou seja, aproximadamente 0.2% do comportamento das variáveis selecionadas, a variável "Total de domicílios Rat%" que é explicado pela variável "Hora Início", mantendo um valor consideravelmente baixo. Além disso, o P-Value foi analisado, que significa a probabilidade de uma hipótese nula ser verdadeira, ou seja, um valor estatístico maior ou igual que os resultados reais observados, determinando se o R-Squared é significativo. Nessa análise, foi verificado o valor do P-value como baixo e arredondado a 0, ou seja, a hipótese nula foi rejeitada.

Lembrando que o Fid% e o Share não foram utilizados, já no teste final, além do "Total de domicílios Rat%", foi inserido os demais Rat% e a precisão caiu para 0.316. Esses testes foram essenciais para verificar a qualidade do modelo e, além disso, foi deduzido que o nível de precisão seria ainda mais baixo com a inclusão do Fid% e Share%. Nesse teste final, foi calculado também o erro quadrático médio (Mean Squared Error), obtendo o valor de 11.083 para teste e 11.092, sendo consideravelmente alto.

Como forma de visualização, foi utilizado o gráfico de correlação como o exemplificado a seguir.



Aqui observa-se correlações menores que 0, ou seja, inversamente relacionadas, ao contrário do valor 1, que possui uma relação alta.

Ademais, também foi testado esse modelo dividindo a Regressão Linear em Share% e Fid%, ambos apresentaram precisão baixa e erro alto que, ao comparar com outros modelos através de gráficos, o resultado vislumbrado foi como o pior modelo a ser usado para predição. Isso se deve ao R2 Score de treino do Share% ser menor que 25% e do teste ser menor que 20%, além do fato do erro absoluto médio de ambos serem extremamente altos.

Como conclusão, não foi escolhido o modelo de regressão linear, pois, esse modelo apresenta uma baixa acuracidade, não correspondendo com o valor real e ajuste ao se tratar das variáveis utilizadas. Por apresentar uma chance de acerto baixíssima e erro quadrático médio alto, é seguido para a aderência de outros tipos de modelos que possuem maior potencial de precisão.

Modelo de Árvore de Decisão - Regressão

Seção de Referência: 5.4.4 Árvore de Decisão.

O modelo de Árvore de Decisão para Regressão mapeia um fluxo de decisões dos possíveis resultados de uma série de escolhas relacionadas. Consiste em uma estrutura de árvore de ponta cabeça, onde cada ramo que é aberto, precisa de uma condição numérica verdadeira ou falsa(no caso, maior, menor, maior ou igual ou menor ou igual um valor), cada uma dessas condições levando a um nó diferente e consequentemente, seguindo caminhos diferentes, sendo a raiz da árvore, a primeira condição do modelo.

Nos testes preliminares, como parte dos tipos de modelos a serem escolhidos, foi treinado e testado o modelo para verificar a sua acuracidade. Nesse caso, para comprovar se esse modelo poderia ser a melhor opção, são seguidos os passos abaixo:

1. Como esperado, foi dividido os valores em variáveis x e y e utilizado estas em treino e teste para para posteriormente realizar a predição utilizando a biblioteca sklearn.
2. A estratégia de escolha para medir a precisão do teste e do treino foi o R-Squared(r2_score), por meio da biblioteca SciKit Learn . Sendo assim, o R-Squared obtido no treino foi de 0.988912842196543, enquanto o do teste foi de 0.9336448126432443. Aqui, considera-se a acurácia como alta, pois, o valor obtido é acima de 0.5(50%) e está próximo de 1.0(100%). Esse resultado foi obtido a partir do uso das features “total de domicílios Rat%".COLOCAR TUDO
3. Não obteve uma medida de acurácia, já que não é possível utilizar o "accuracy_score" para modelos de regressão.
4. Nesse teste final, calcula-se também o erro quadrático médio (Mean Squared Error), obtendo o valor de 0.16004108534213854 para treinamento e 0.9639167769744844 para testes, sendo ambos consideravelmente baixos, mas explicitando um possível problema de overfitting, já que o resultado para treinamento foi muito bom(próximo de 0) e o de teste se mostrou muito divergente do valor de treinamento, podendo indicar que os dados se viciaram ao dataset de teste e acabam não tendo um bom desempenho ao receber novos dados(dataset de treinamento).

Como conclusão, não foi escolhido o modelo de árvore de decisão em referência a todas as features, pois por mais que o modelo esteja elencado em terceiro maior na precisão em relação aos outros testados, há uma forte hipótese de que esteja acontecendo overfitting na predição, o que faz com que a Árvore de Decisão não corresponda tão bem ao receber novos dados.

Otimização dos hiperparâmetros da Árvore de Decisão

Após o algoritmo do modelo de árvore de decisão ser treinado, os hiperparâmetros foram ajustados, como o max_depth(profundidade máxima), que indica o quão profunda a árvore é, ou seja, quanto mais profunda, mais divisões a árvore possui, fazendo ter uma maior captura de informações sobre os dados, e o min_samples_leaf(folha de amostras mínimas), usada para representar o número mínimo de amostras adequadas para estar em um nó folha. Nesse contexto, tanto a profundidade máxima quanto a divisão mínima da amostra podem prevenir o overfitting. Para avaliar o modelo, foi usado a validação cruzada que permite dividir o conjunto em várias partes, garantindo que o modelo tenha uma boa performance com o pacote Scikit-learn, servindo como meio de recursividade automática. Ao verificar a performance, as colunas foram analisadas separadamente, sendo Share%, Rat% e Fid%, obtendo a precisão e o MSE (Erro Quadrático Médio) de cada coluna com o uso do Grid Search (Pesquisa em grade), uma técnica de treinamento do modelo e de diferentes valores de hiperparâmetros, a qual são passadas combinações. É a partir dessa técnica que é fornecido o cálculo do erro, além de possibilitar a escolha dos melhores valores.

Sendo assim, foi visualizado e comparado dados de teste e de treino e, ambas colunas apresentaram valores de precisão de teste inferiores ao de treino, como pode ser visto nas tabelas a seguir:

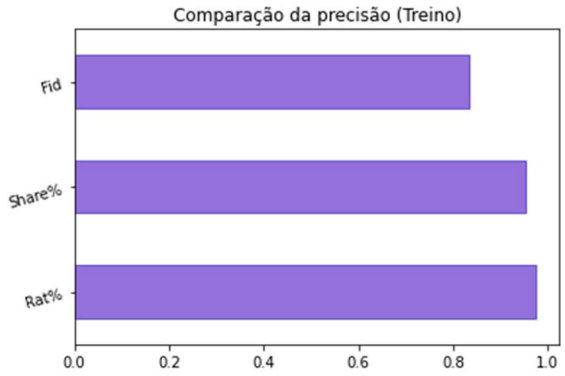


Tabela 1. Precisão das colunas de treino

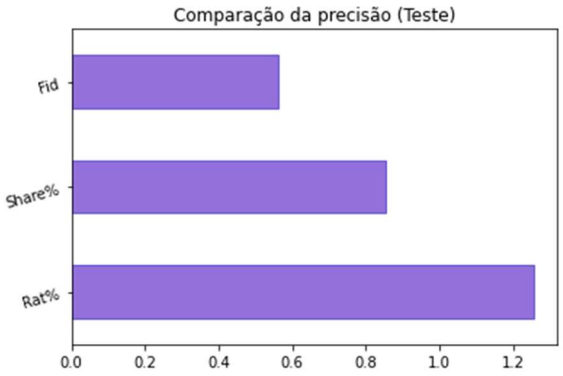


Tabela 2. Precisão das colunas de teste

Nessa tabela, é possível visualizar os dados exatos de precisão do treino e teste das colunas Rat%, Share% e Fid%, sendo nítido os valores inferiores do teste em comparação com o treino. Lembrando que, a métrica utilizada para calcular a precisão foi o R2 Score.

Tabela - Precisão dos dados Árvore de Decisão

	Treino	Teste
Rat%	0.9761	0.9127
Share%	0.9545	0.8532
Fid%	0.8331	0.5628

Vale salientar discrepâncias, já que o teste apresentou valores mais baixos que o treino, além do erro tanto no treino, quanto no teste, foram altos, principalmente ao se tratar do Fid%. Como todos os hiperparâmetros tiveram erros altos, foi descoberto a presença de overfitting e, para reverter essa situação, foi buscado alternativas para minimizar o sobreajuste dos dados, como alterar o valor da profundidade máxima, diminuindo e aumentando com valores altos, sendo estabelecido com o valor fixo de 2000, a divisão mínima da amostra sempre acima de 2, tendo o valor estabelecido de 2 e a poda da árvore. As primeiras tentativas foram alterar o valor da profundidade e a divisão mínima, porém, não houve diferenças significativas comparado aos resultados anteriores, muito pelo contrário, houve até mesmo uma piora nos resultados.

Modelo Random Forest - Regressão

Seção de referência: 5.4.3 Random Forest Regressor

Random Forest, por tradução direta significa Floresta Aleatória, fazendo referência proposital ao modelo testado anteriormente, Árvore de Decisão. Nesse caso, ao ter um modelo que combina a lógica das árvores de decisão com a flexibilidade e aleatoriedade de seleção de atributos, ao invés da seleção a partir do cálculo de impureza, então a precisão final dos resultados seriam melhorados. Primeiramente, assim como outras alternativas de modelos, foi testado sua precisão, e nele foi obtido os dados a seguir.

Calculando com a medida Rat%, foi obtido como precisão em treinamento uma taxa de 0.984914874246196 (que significa aproximadamente 98%), enquanto o de teste resultou em 0.9452038593500062 (que significa aproximadamente 94%). Tendo como erro de treino e teste, respectivamente, 0.27969560045998526 e 0.6821303900067482.

Já calculando com o Share%, foi encontrado 0.9652199665107021(que significa aproximadamente 96%) na precisão de treino, e no teste 0.8947784930237088 (que significa aproximadamente 89%). Tendo como erro de treino e teste, respectivamente, 15.95812322126885 e 37.2333956223377.

Com o Fid%, ficou em 0.8539301670715912 (que significa aproximadamente 85%) referente ao treino e com 0.6110654322602026 (que significa aproximadamente 61%) referente ao teste de precisão. Tendo como erro de treinamento e de teste, respectivamente, 82.76312386603972 e 182.46552387420184.

Após isso, procura-se encontrar as Features com maior importância para cada medida, e foram encontradas com mais destaque, respectivamente:

- Rat%: Hora Início, Novela, Dia, Dia da Semana e Mês.
- Share%: Dia, Hora Início, Dia da Semana e Mês.
- Fid%: Dia, Hora Início, Dia da Semana e Mês.

Como conclusão, foi escolhido o modelo de Random Forest referente a todas as features, pois os resultados obtidos com o modelo resultou em uma confiança nos dados, visto que estes estão de acordo com o esperado, e assim, culminando para não ocorrência de overfitting, que ocorreria caso fosse utilizado a árvore de decisão como modelo principal.

Como a aprendizagem do modelo Random Forest é composta por diversas Árvores de Decisão, seus hiperparâmetros são praticamente os mesmos. Para a tarefa de otimização do modelo, foi utilizado o método Random Search. O Random Search segue o mesmo princípio do Grid Search, prover uma malha de parâmetros e para cada um deles um valor ou intervalo. Após esse processo, o sistema pega os valores fornecidos e/ou escolhe aleatoriamente valores presentes no intervalo definido e testa todas as combinações possíveis entre eles. Os valores que apresentarem melhores resultados são retornados.

Os valores anteriores aos testes de hiperparâmetros foram:

```
## Rat %

Precisão (treino):  0.984914874246196

Mean Squared Error (Treino):  0.21784669918708263

Precisão (teste):  0.9452038593500062

Mean Squared Error (teste):  0.7964637087271896
```

Share %

Precisão (treino): 0.9652199665107021

Mean Squared Error (Treino): 11.795810564160996

Precisão (teste): 0.8947784930237088

Mean Squared Error (teste): 35.54252553971531

Fid %

Precisão (treino): 0.8539301670715912

Mean Squared Error (Treino): 63.68058045962253

Precisão (teste): 0.6110654322602026

Mean Squared Error (teste): 173.19073638784016

Para a realização do primeiro teste, foram definidos “valores fantasmas”. Intervalos e valores menores. Após o teste os resultados foram:

Rat %

Precisão (treino): 0.7187704708241583

Mean Squared Error (Treino): 4.037364561963939

Precisão (teste): 0.7036470744951845

Mean Squared Error (teste): 4.277266094890892

Share %

Precisão (treino): 0.4070410154893055

Mean Squared Error (Treino): 201.5258232324657

Precisão (teste): 0.38492621604197425

Mean Squared Error (teste): 208.60833653132866

Fid %

Precisão (treino): 0.33696309612527425

Mean Squared Error (Treino): 456.2219701549914

Precisão (teste): 0.30254069694966607

Mean Squared Error (teste): 477.036765254523

Pode-se observar que os resultados depois da tentativa de otimização de hiperparâmetro foram inferiores a versão piloto do modelo. Por consequência, para um segundo teste, os intervalos foram espaçados consideravelmente para que se obtivesse um maior número de combinações e maior probabilidade de acurácia. Porém, os resultados não foram satisfatórios, sendo eles:

```
## Rat %

Precisão (treino):  0.7460230644410962

Mean Squared Error (Treino):  3.655179363526605

Precisão (teste):  0.7290699881126778

Mean Squared Error (teste):  3.918069233211218

## Share %

Precisão (treino):  0.40835540037648715

Mean Squared Error (Treino):  201.02089769009805

Precisão (teste):  0.3858609325841572

Mean Squared Error (teste):  208.23090819062836

## Fid %

Precisão (treino):  0.33897648340402725

Mean Squared Error (Treino):  455.0714483406486

Precisão (teste):  0.30443323770808567

Mean Squared Error (teste):  475.9471187364417
```

Tabela - Melhores resultados de precisão dos dados do modelo Random Forest.

	Treino	Teste
Rat%	0.9849	0.9452
Share%	0.9652	0.8947
Fid%	0.8539	0.6110

Modelo LightGBM - Regressão

Célula de Referência: 5.4.5 LightGBM Rat%

Light GBM é um algoritmo de Gradient Boosting, otimizado, do tipo ensemble. Sua estrutura consiste em uma série de modelos "fracos" de árvores de decisão, combinados, como uma espécie de "comitê" de escolha, que então, a partir da média dos resultados, gera uma previsão. Sua principal diferença, quando comparado ao método Random Forest, é que, durante o processo de aprendizado, as árvores de decisão crescem de acordo com a direção de suas "folhas" (leaf-wise), ao invés de expandir-se pelo nível dos nós (level-wise). O rápido aprendizado e baixo uso de memória constituem parte dos vários benefícios deste modelo. Em contrapartida, este é consideravelmente mais sensível ao Overfitting, em datasets com baixo número de linhas.

Durante o processo de treinamento do modelo, foi necessária a modificação de alguns hiperparâmetros, como taxa de aprendizado e número de estimadores/iterações. O primeiro, foi definido com o valor padrão de 0.9, presente na documentação do mesmo. O segundo, foi escolhido a partir de testes, começando com apenas 100 estimadores e incrementando o número de iterações ao longo do teste.

A medida em que o parâmetro "n_estimators" foi aumentado, o tempo de treino do modelo cresceu linearmente, da mesma forma, impactando positivamente a acurácia resultante.

Após 30.000 iterações, obteve-se um R2 Score de aproximadamente 0.98, com um MSE (Erro quadrático Médio) de 0.82. Constatou-se uma alta performance e acurácia com o uso deste modelo. Entretanto, devido à limitações, foi possível a aplicação apenas para o rótulo "Total Domicílios I Rat%", sendo requerida a replicação para cada uma das colunas esperadas, no conjunto target, de modo individual.

Com o propósito de aplicar uma maior complexidade ao modelo LightGBM, foi-se utilizado o pacote optuna, uma estrutura de software de otimização automática de hiperparâmetros. Além disso, foi implementada a ferramenta verstack, que automatiza o próprio optuna, tornando o processo mais rápido e detalhado para o operador do modelo. O verstack aplicado ao LightGBM, permite ajustar hiperparâmetros por meio de uma integração utilizando LightGBMTuner. Esta função é responsável por implementar uma estratégia de ajuste de hiperparâmetro adequada para o modelo. Os parâmetros abaixo foram identificados em ordem de prioridade para o modelo:

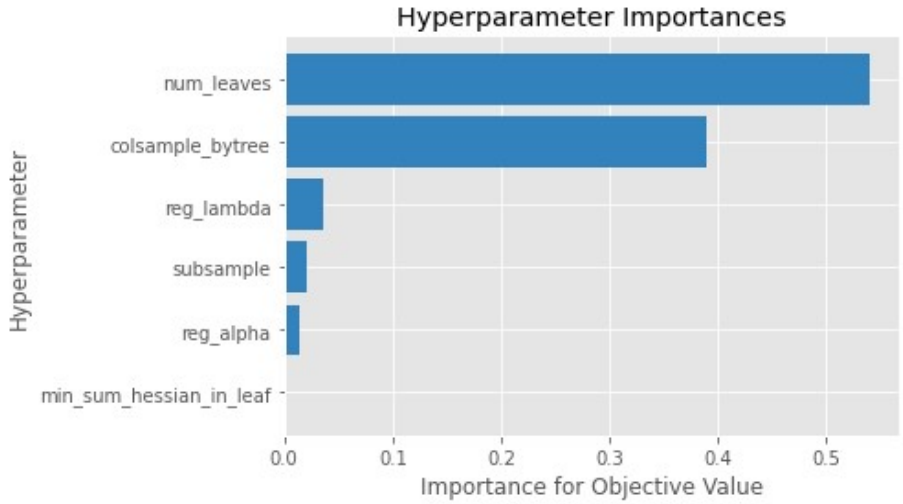


Gráfico X - ordem de prioridade dos hiperparâmetros do modelo.
Fonte: Criação própria (2022).

O num_leaves é o parâmetro mais importante no modelo LightGBM, é responsável por determinar o número de folhas em uma única árvore de decisão. A folha de decisão é o local onde é realizada a decisão real. Por essa razão, o impacto do num_leaves no aprendizado do modelo é superior ao max_depth.

O valor de max_depth é responsável pelos níveis das árvores, tempo de treinamento e performance. No modelo, o valor de max_depth foi definido como none, assim, os nós da árvore são expandidos até que todas as folhas sejam puras, elas apenas terão rótulos com a definição do valor de min_samples_leaf para o padrão um. Com o propósito de alcançar uma maior acurácia para o modelo, foram aplicados uma combinação dos parâmetros n_estimators e learning_rate, que controlam o número de árvores de decisões e o tempo de aprendizagem do modelo.

Além disso, também foram aplicados parâmetros de regularização. A função min_sum_hessian_in_leaf indica a soma de hessianos dos pontos de dados na folha da árvore por meio de uma matriz quadrada de derivadas parciais de segunda ordem de uma função de valor escala. O cálculo da hessiana de um ponto de dados é feito com a derivada de segunda ordem da loss function w.r.t. e o valor de previsão atual, onde se é possível visualizar como cada derivada parcial afeta as demais derivadas parciais. No momento em que se realiza a divisão de estimadores m=1,..M (árvores/estimadores), se a soma do hessiana em uma folha for menor que a função, a árvore para de crescer. Em regressão, a loss function é definida por loss(y, pred) = 1/2 * (pred - y)^2, nesse caso a hessiana é equivalente ao número de pontos da dados na folha. No projeto, o valor utilizado para essa função foi próximo a zero, assim, as árvores crescem sem restrições e se ajustam às demais existentes.

O parâmetro subsample especifica a quantidade de amostras selecionadas que serão utilizadas por iteração de construção de árvores, isso aprimora a velocidade de

treinamento do modelo visto que algumas linhas serão selecionadas aleatoriamente para ajustar-se em cada árvore. O colsample_bytree possui a mesma função que o subsample, mas representa a razão de amostras de colunas ao construir cada árvore. Nesse parâmetro a subamostragem ocorre uma vez por cada árvore construída e é utilizada para realizar o treinamento das features do modelo. O reg_aplha é um cognome do lambda_l1, um parâmetro aplicável para evitar o overfitting por meio da funcionalidade da penalização de features encontradas que não aumentam a precisão do modelo. Ademais, a função reg_lambda, um cognome de lambda_l1, também é aplicável para evitar o overfitting.

Para avaliação dos hiperparâmetros do modelo, aplicou-se a métrica MSE (Erro Quadrático Médio). Por padrão, o LightGBM executou 100 testes para tentativas de treinamento/validação dos parâmetros selecionados aleatoriamente no espaço de busca.

Na tabela abaixo, são apresentados os resultados obtidos para teste e treino do modelo, para as colunas de Rat%, Fid% e Shr%. A coluna de Fid% se manteve com um menor valor que as demais colunas.

Tabela - Precisão dos dados LightGBM

	Treino	Teste
Rat%	0.98	0.956
Share%	0.95	0.889
Fid%	0.60	0.524

Modelo K-Nearest Neighbors - Regressão

Seção de referência: 5.4.2 KNN Regressor

O algoritmo KNN usa a "semelhança entre dados vizinhos" para prever os valores de quaisquer novos pontos de dados. Assim, esse novo ponto recebe um valor que tem relação com os valores base do conjunto de dados semelhantes utilizados no treinamento. O KNN também é um algoritmo de aprendizado de máquina supervisionado, o modelo se baseia em dados de entrada rotulados para aprender uma função que produz uma saída apropriada quando recebe novos dados não rotulados. Sendo que essas variáveis são definidas como dependentes "y", e variáveis independentes "x". Uma segunda propriedade desse modelo, é que ele não é linear, logo, pode-se representar os resultados graficamente de diferentes modos.

Para o treinamento do modelo, foi criado três diferentes valores de outputs, o rat%, share% e fid% (y), assim foi definido três modelos e foi utilizado o mesmo conjunto de features como input (x) em cada modelo. Ademais, visando diminuir o erro de previsão do valor K, para todos os outputs, foi utilizado o cálculo de RSME (Root-mean-square deviation), considerando um valor máximo de 20. O RMSE corresponde à raiz quadrada da diferença média entre os valores de resultados conhecidos observados e os valores previstos, RMSE = média((observados - previstos)^2) %>% sqrt(). Logo, quanto menor o RMSE, melhor o modelo, e para os outputs de share%, fid% e rat%, o melhor valor encontrado foi K= 3.

Para testar a precisão do modelo, foi utilizado o cálculo R-Squared(r2_score) e para verificar a média de erro do modelo, foi aplicado o cálculo de Mean Squared Error. Portanto, após a realização dos testes, foi obtido o seguinte resultado:

```
## Rat %
Precisão (teste):  0.9533961572465708
Mean Squared Error (teste):  0.6821303900067482

## Share %
Precisão (teste):  0.8890592799933043
Mean Squared Error (teste):  37.2333956223377

## Fid %
Precisão (teste):  0.6019585416175695
Mean Squared Error (teste):  182.46552387420184
```

Com os resultados,constata-se que, para os valores de Rat%,o modelo obteve uma excelente taxa de precisão, com os valores se aproximando de 1, indicando que o modelo chegou bem próximo dos conjuntos de dados do treinamento. Entretanto, para os valores de outputs do Share% e Fid%, o modelo apresentou valores elevados no Mean

Squared Error, demonstrando que a acurácia dos testes executados talvez não estejam próximos dos valores esperados.

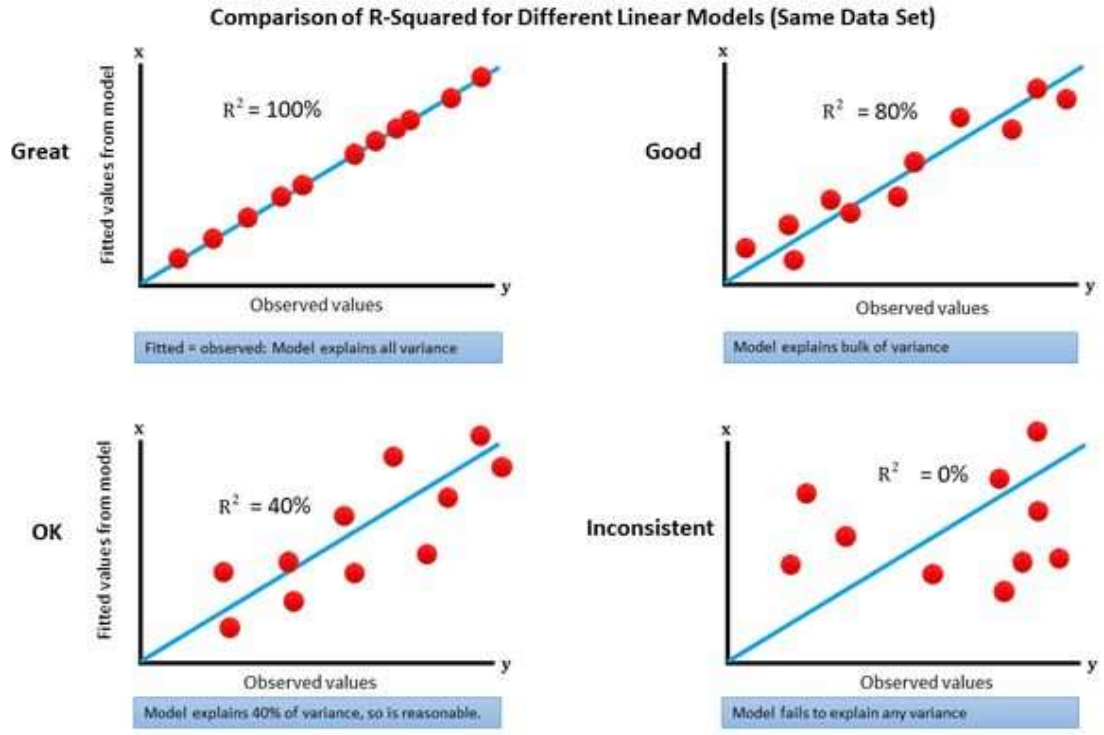
4.5. Avaliação

Para a avaliação e comparação dos resultados obtidos nos modelos testados, foram utilizadas duas métricas principais: R-Squared Score(R^2 Score) e MSE (Erro quadrático médio).

R-Squared Score

$$R^2 = \frac{SSR}{SST} = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

Esta métrica é uma medida estatística que mede a força de correlação entre uma variável dependente e independente, variando de 0 (pouco relacionado) a 1 (muito relacionado). Com ela, entende-se a explicabilidade de uma saída, a partir das features de entrada. Considerando a sua variação, é possível entender seus resultados de forma intuitiva, facilitando a comparação dos modelos.

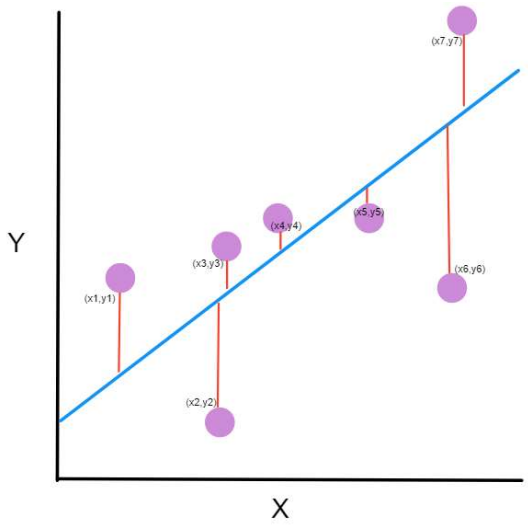


Fonte: https://www.datasciencecentral.com/r-squared-in-one-picture/

MSE (Mean Squared Error)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Elaborada por Karl Pearson, o erro quadrático médio, ou “Mean Squared Error”, compara a distância entre os valores resultantes da predição de um modelo, com o valor real esperado, ao quadrado. Quanto menor a medida, maior a qualidade do modelo, e mais próximo da linha de regressão, estão os pontos resultantes. Define-se então, o menor valor, como ponto de maior acurácia. Esta métrica foi aplicada pela sua sensibilidade à altos erros.



Fonte: https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/

4.5.1 Avaliação dos Resultados (sem aplicação de otimização dos hiperparâmetros)

Após um extenso processo de testes, com múltiplas iterações, os seguintes resultados foram obtidos para os modelos:

Treino Coluna Rat%

Modelo	R2_Score (Treino)	MSE (Treino)	Série de Colunas
Regressão Linear	0.32	11.08	Rat%
KNN Regressor	0.98	0.27	Rat%
Random Forest Regressor	0.98	0.22	Rat%
Árvore de Decisão	0.99	0.16	Rat%

Tabela x - Resultados de treino coluna Rat%, com as métricas MSE e R2 dos modelos.

Treino Coluna Shr%

Modelo	R2_Score (Treino)	MSE (Treino)	Série de Colunas
Regressão Linear	0.11	291.5	Shr%
KNN Regressor	0.95	15.9	Shr%
Random Forest Regressor	0.96	11.8	Shr%
Árvore de Decisão	0.97	10.0	Shr%

Tabela x - Resultados de treino coluna Shr%, com as métricas MSE e R2 dos modelos.

Treino Coluna Fid%

Modelo	R2_Score (Treino)	MSE (Treino)	Série de Colunas
Regressão Linear	0.06	694.0	Fid%
KNN Regressor	0.82	82.8	Fid%
Random Forest Regressor	0.85	63.7	Fid%
Árvore de Decisão	0.86	57.9	Fid%

Tabela x - Resultados de treino coluna Fid%, com as métricas MSE e R2 dos modelos.

Para os resultados de treino da coluna de Rat%, Fid% e Shr%, o modelo de regressão linear foi o modelo que obteve a pior precisão e o seu MSE demonstra uma taxa alta de erro, aumentando principalmente para a coluna de Fid%. Em termos da coluna Shr%, os demais modelos demonstraram um crescimento na taxa do MSE, mas não apresentaram uma queda significativa em suas performances na métrica do R2. Contudo, para a coluna de Fid% todos os modelos tiveram uma elevada redução da precisão do R2 e um elevado salto no MSE.

Teste Coluna Rat%

Modelo	R2_Score (Teste)	MSE (Teste)	Série de Colunas
Regressão Linear	0.32	11.09	Rat%
KNN Regressor	0.95	0.68	Rat%
Random Forest Regressor	0.94	0.80	Rat%
Árvore de Decisão	0.93	0.94	Rat%

Tabela x - Resultados de teste coluna Rat%, com as métricas MSE e R2 dos modelos.

Teste Coluna Shr%

Modelo	R2_Score (Teste)	MSE (Teste)	Série de Colunas
Regressão Linear	0.11	290.3	Shr%
KNN Regressor	0.89	37.2	Shr%
Random Forest Regressor	0.89	35.5	Shr%
Árvore de Decisão	0.87	42.0	Shr%

Tabela x - Resultados de teste coluna Shr%, com as métricas MSE e R2 dos modelos.

Teste Coluna Fid%

Modelo	R2_Score (Teste)	MSE (Teste)	Série de Colunas
Regressão Linear	0.06	693.53	Fid%
KNN Regressor	0.60	182.5	Fid%

Random Forest Regressor	0.61	173.2	Fid%
Árvore de Decisão	0.53	210.70	Fid%

Tabela x - Resultados de teste coluna Fid%, com as métricas MSE e R2 dos modelos.

Para teste, o modelo de regressão linear também apresentou a pior performance em relação às métricas de avaliação utilizadas. Para a coluna Shr%, oposto ao treino, os demais modelos demonstraram um crescimento significativo tanto na taxa do MSE e R2. Ademais, para a coluna de Fid% todos os modelos tiveram uma elevada redução da precisão do R2 e um elevado salto no MSE, em níveis que não foram observados no treino.

Comparação Rat% R2 Score

Seção 5.4.6.1 Score

Utilizando a métrica R2 Score para calcular a precisão do Rat%, Shr% e Fid%, as tabelas abaixo representam a comparação dos resultados de treino e teste de cada modelo:

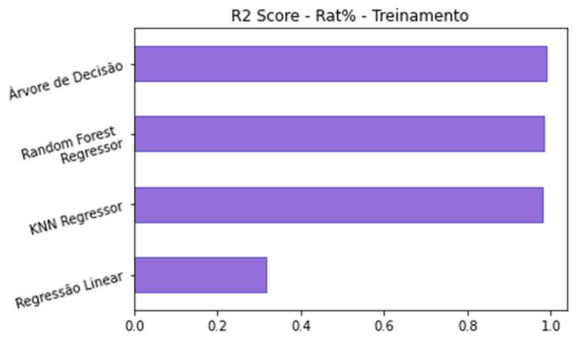


Tabela x - R2 Score - Rat% - Treinamento.

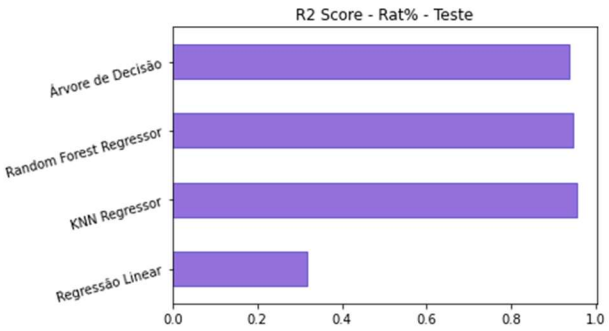
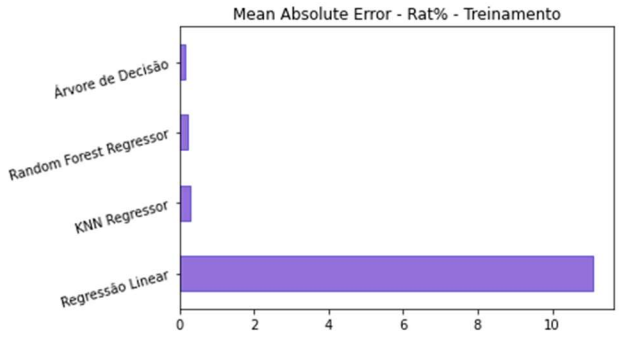


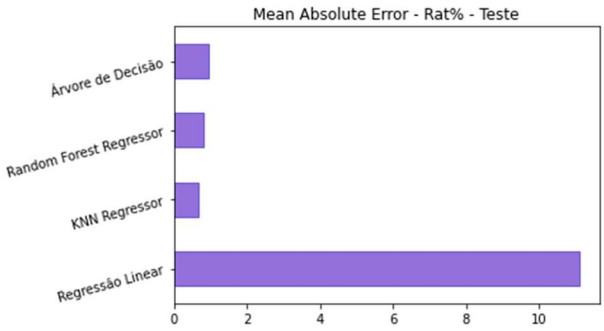
Tabela x - R2 Score - Rat% - Teste

Comparação Rat% MAE

Seção 5.4.6.2 Mean Absolute Error Score

Aqui é comparado o Erro Absoluto Médio de treino e teste de cada modelo, sendo o modelo de Regressão Linear Múltipla com o valor mais elevado, havendo uma discrepância.

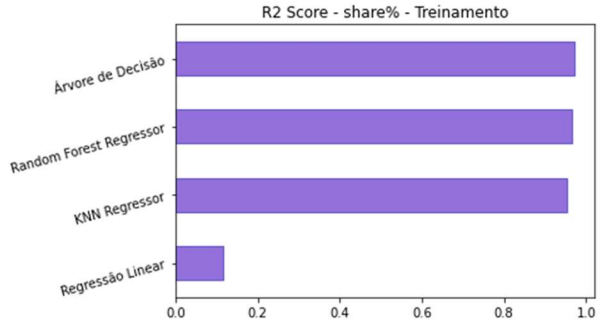


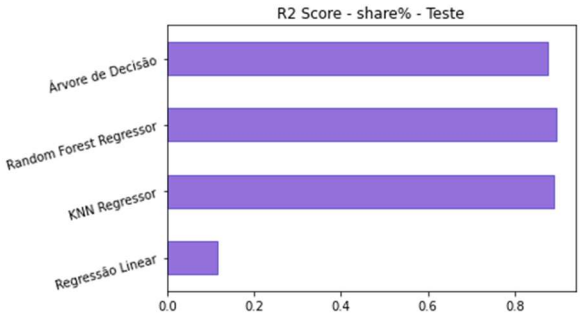


Comparação Share% R2 Score

Seção 5.4.7.1 R2 Score

Dessa vez, é tratado a comparação do Share% de treinamento e teste dos modelos por meio do R2 Score, como pode-se ver a seguir:

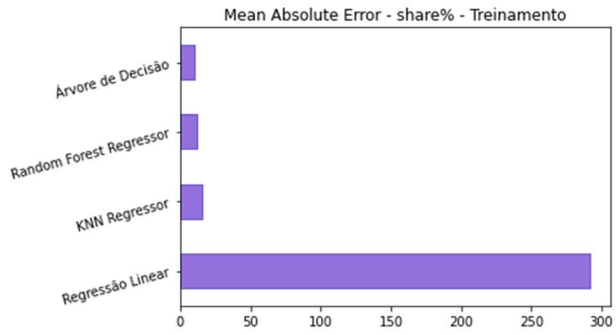


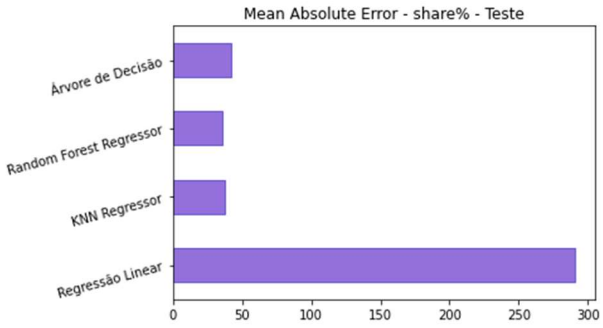


Comparação Share% MAE

Seção 5.4.7.2 Mean Absolute Error Score

Aqui, observa-se a comparação do Erro Absoluto Médio dos modelos tratando-se do Shr%, de treino e teste. A primeira tabela é a comparação dos modelos em relação aos resultados de treino e a segunda tabela são os resultados de teste de cada modelo:





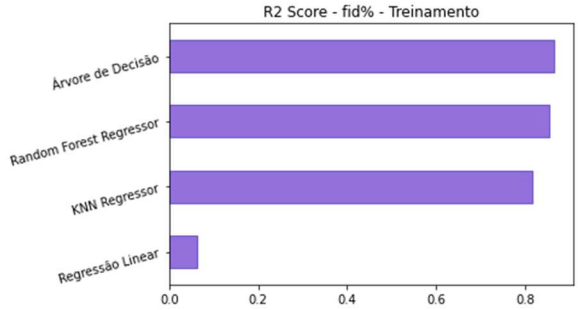
Fonte: Matplotlib Colab Data Oracles

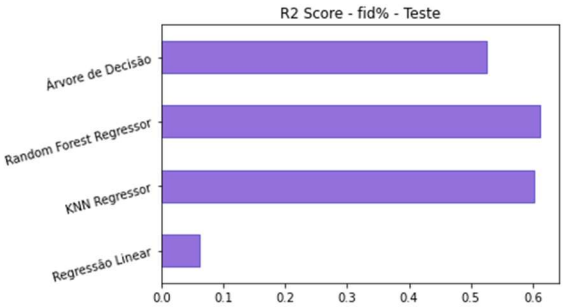
Fonte: Matplotlib Colab Data Oracles

Comparação Fid% R2 Score

Seção 5.4.8.1 R2 Score

As tabelas a seguir trata-se dos resultados do Fid% de cada modelo, tendo a comparação dos resultados de treino na primeira tabela e a comparação dos resultados de teste na segunda tabela:





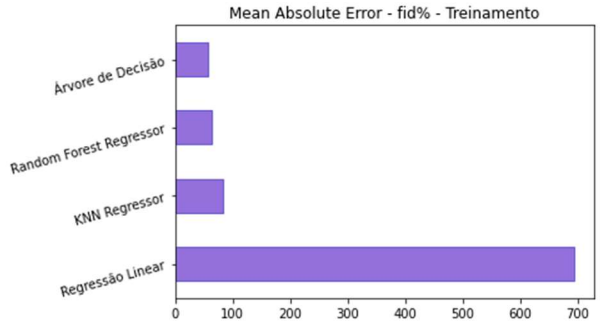
Fonte: Matplotlib Colab Data Oracles

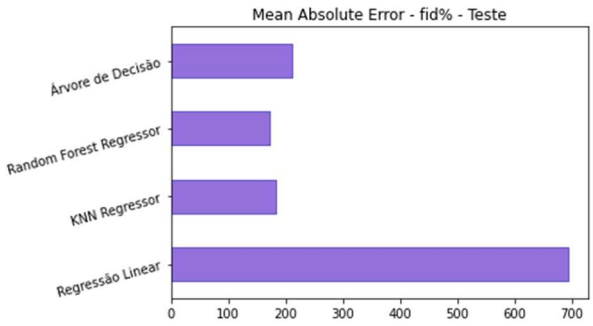
Fonte: Matplotlib Colab Data Oracles

Comparação Fid% MAE

Seção 5.4.8.2 Mean Absolute Error Score

O Erro Médio Absoluto a seguir é comparado entre todos os modelos de treino e teste, levando em consideração o fid%. A primeira tabela trata-se da comparação dos modelos referente ao treino, enquanto a segunda tabela trata-se do teste:





Fonte: Matplotlib Colab Data Oracles

4.5.2 Avaliação dos Resultados (com aplicação de otimização dos hiperparâmetros)

Modelo de Árvore de Decisão Regressiva

Diante de pesquisas efetuadas, houve uma busca de alternativas para melhorar o péssimo desempenho e para solucionar o problema do overfitting, tal como a poda da árvore de decisão ou, em outras palavras, descartar qualquer divisão que não agregue valor significativo ao modelo. No entanto, como houve uma redução imensa para deixar o modelo o mais simples possível, a pré poda, que envolve cortar a árvore depois de construída e super adaptá-la, não seria viável e nem uma boa opção. Desse modo, foi vislumbrado que a árvore de decisão é um dos modelos de machine learning mais suscetíveis ao overfitting e é mais adequada para se trabalhar com modelos simples, o que não é o caso. Além disso, pelo fato de trabalhar com uma ampla variedade e quantidade de dados, considerando o modelo como complexo e não simples, o crescimento de uma árvore até um certo nível torna o overfitting inevitável, pois quanto mais complexo, mais propício é o resultado de sobreajuste e expansão horizontal ao invés do vertical (expansão mais adequada) da árvore. As tabelas a seguir permitem uma melhor visualização referente a precisão e erro em comparação dos dados de treino e de teste das colunas.

Aqui percebe-se que o MSE aumentou significativamente para todas as colunas e apresentou um valor superior à precisão. Em comparação com o Fid% de treino, por exemplo, o Fid% pulou de 74 para 210 no de teste, sendo um valor extremamente alto.

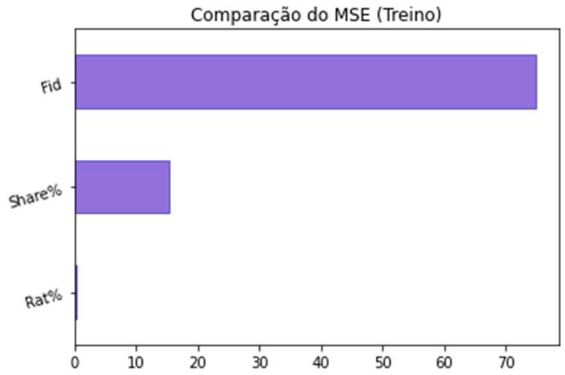


Tabela 3. Valor do MSE das colunas de treino

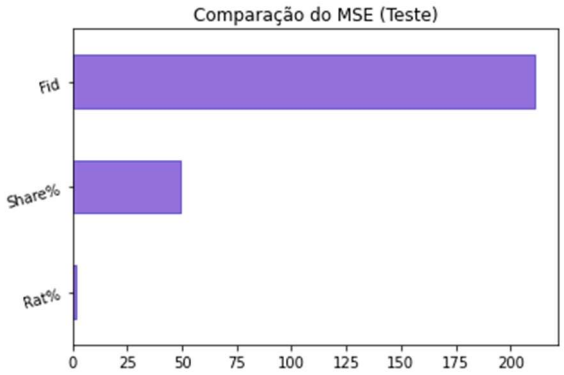


Tabela 4. Valor do MSE das colunas de teste

Na tabela abaixo, é possível visualizar os resultados e a taxa de erro com mais exatidão, comparando o MSE (Erro Médio Quadrático) de todas as colunas em relação ao treino e teste.

Tabela - Erro Médio Quadrático Árvore de Decisão

	Treino	Teste

Rat%	0.3413	1.2591
Share%	15.3387	49.4119
Fid%	74.9612	210.9565

Através da tabela, é vislumbrado que o erro de teste ultrapassou os valores de erro do treino de maneira significativa.

As próximas tabela é uma comparação entre o MSE e a precisão com o R2 Score, onde o MSE ultrapassou a precisão, sendo que o mais adequado é o MSE ser inferior aos valores de precisão para obter uma predição confiável e com qualidade. Lembrando que os valores de teste são os mais relevantes e são os valores válidos para a predição, pois são valores desconhecidos.

Tabela - Comparação MSE e Precisão (Treino)

	Precisão(treino)	MSE(treino)
Rat%	0.9761	0.3413
Share%	0.9545	15.3387
Fid%	0.8331	74.9612

Tabela - Comparação MSE e Precisão (Teste)

	Precisão(teste)	MSE(teste)
Rat%	0.9127	1.2591
Share%	0.8532	49.4119
Fid%	0.5628	210.9565

Através dessa visualização mais exata da comparação entre o MSE e a Precisão de cada coluna, é observado o quão superior é o MSE em relação a precisão.

Devido a esses resultados e ao longo tempo dedicado para a resolução do problema de overfitting, o modelo de árvore de decisão regressiva foi descartado pelo fato de se encaixar melhor em modelos de dados simples e pelo grande risco de não apresentar boas predições referente a novos dados, pois, não há adequação suficiente para alcançar o objetivo de uma boa predição sem enviesamento, já que o teste obteve resultados menores que o treino e o erro foi superior a precisão. Ou seja, não é o melhor modelo para se trabalhar, sendo o pior em comparação aos outros modelos que tiveram testes de hiperparâmetros realizados.

Modelo LightGBM

Analisando os resultados obtidos para a métrica MSE no modelo, é possível constatar que a taxa de erro para a coluna Rat% não apresentou um valor tão discrepante. Entretanto, para a coluna de Shr%, foi observado um aumento de 21 pontos entre o treino e teste, e para a coluna Fid%, a taxa de erro se manteve em 200 pontos, com diferença de 44 pontos entre treino e teste.

Tabela - Erro Médio Quadrático LightGBM

	Treino	Teste
Rat%	0.275	0.634
Shr%	16.731	37.207
Fid%	254.697	298.586

Ademais, comparando as tabelas de MSE e Precisão, para os treino e testes das colunas, é constatável que o elevado valor em MSE para Fid%, reflete que o modelo não está adequado para predição de valores dessa coluna, o valor de teste foi de 52%. Todavia, para os valores de Shr% e Rat%, o MSE obtido pelo modelo, pondera que o modelo está adequado para predição, principalmente levando-se em consideração os baixos valores obtidos para a coluna de Rat%.

Tabela - Comparação MSE e Precisão (Treino)

	Precisão(treino)	MSE(treino)
Rat%	0.98	0.275
Share%	0.95	16.731
Fid%	0.60	254.697

Tabela - Comparação MSE e Precisão (Teste)

	Precisão(teste)	MSE(teste)
Rat%	0.956	0.634
Share%	0.889	37.207
Fid%	0.524	298.586

Desse modo, o modelo LightGBM, por não apresentar uma diferença de valores tão divergentes entre teste e treino para as colunas, é apto de utilização como principal modelo para o presente projeto, visto que os resultados obtidos para as colunas Rat% e Shr% apresentam uma alta precisão e baixa taxa de MSE. Contudo, o valor de MSE de Shr% ainda é considerado um valor destoante do que se é considerado adequado, mas levando-se em consideração as colunas utilizados como output do modelo (todas colunas de Shr% presentes no dataset), é possível constatar que manipulações podem ser realizadas com a finalidade de diminuir esse valor e obter-se um modelo mais preciso.

Modelo K-Nearest Neighor (KNN)

Dado que o modelo "K-Nearest Neighbor" (KNN) já está treinado, para a utilização dos hiperparâmetros foi usado a biblioteca ScikitLearn, e assim tendo a possibilidade de manusear métodos que auxiliam na sistematização da busca pelos hiperparâmetros mais adequados para o modelo. Com isso, foram utilizados métodos como GridSearch e RandomSearch que são práticas para treinamento do modelo sendo combinados com diferentes hiperparâmetros. Desse modo, para uma melhor análise de performance foram utilizados os seguintes hiperparâmetros:

- n_neighbors: total de vizinhos que estão padronizados para k-neighbors.
  parameters = {'n_neighors': range(2,100)}
- weights: foi utilizado para medir uma predição sobre os seguintes pesos

uniform: há uma uniformidade na ponderação nos pontos em cada vizinhança.

- distance: é a medição entre pontos, concluindo que quanto mais distantes menos influência haverá nos dados.

parameters = {'weigths:' ['uniform', 'distance']}

Comparações de resultado Rat% com e sem weight distance, respectivamente:
Rat% com weight
Precisão (treino): 0.9982567862021395
Mean Squared Error (treino): 0.25147001845238104
Precisão (teste): 0.7482785060602518
Mean Squared Error (teste): 3.6639074965676124

Rat% sem weight
Precisão (treino): 0.86860355589410084
Mean Squared Error (treino): 1.9598440434126871
Precisão (teste): 0.688555541827056
Mean Squared Error (teste): 4.541871749960316

Comparações de resultado Share% com e sem weight distance, respectivamente:
Share % com weight
Precisão (treino): 0.9956248417172455
Mean Squared Error (treino): 1.4612731081547603
Precisão (teste): 0.5025707295930427
Mean Squared Error (teste): 168.96453682529543

Share % sem weight
Precisão (treino): 0.8686035589410084
Mean Squared Error (treino): 1.959844043412687
Precisão (teste): 0.39552429527975563
Mean Squared Error (teste): 205.3054631895634

Comparações de resultado Fid% com e sem weight distance, respectivamente:
Fid % com weight
Precisão (treino): 0.979534934037132
Mean Squared Error (treino): 9.082970577261907
Precisão (teste): 0.23368215185794664
Mean Squared Error (teste): 479.7348019061449

Fid % sem weight
Precisão (treino): 0.6274369098506881
Mean Squared Error (treino): 233.3624367893646
Precisão (teste): 0.177506752519112199
Mean Squared Error (teste): 544.3814773169906

- callable: retorna uma mesma matriz que aceita as distâncias dos vizinhos.
- algorithm: para calcular os vizinhos mais próximos usamos esse parâmetros, acrescentando o "brute" o algoritmo se baseia na apropriação dos valores relacionados ao método do fit.

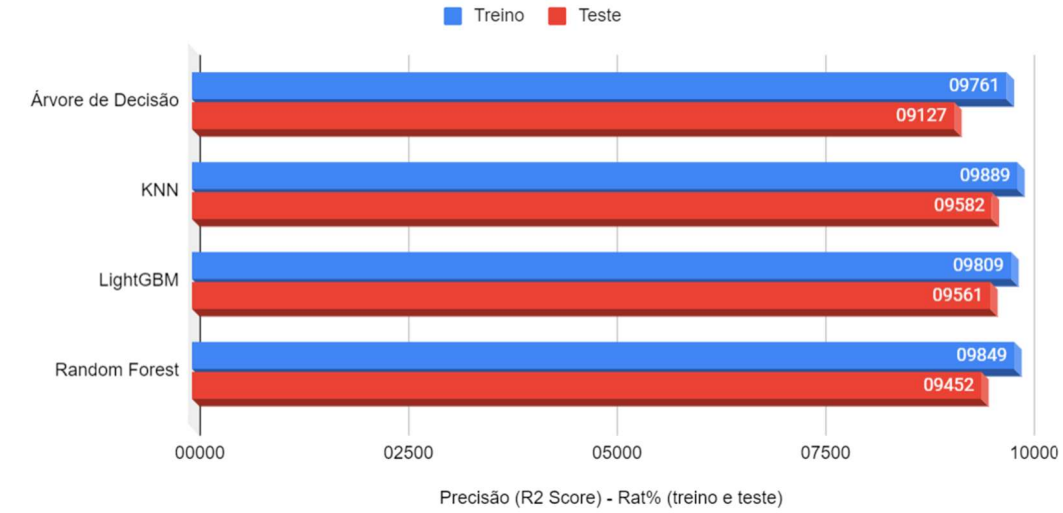
parameters = {'algorithm': ['brute', 'kd_tree', 'ball_tree'}

Diante desse cenário, é notória a participação positiva dos métodos utilizados, ficando evidente que tanto o erro de teste quanto no treino os valores foram relativamente menores quando aplicados aos métodos, tendo dessa maneira uma maior precisão quando usado. Em destaque, o Fid% teve um melhor resultado no quesito Mean Squared Error (erro quadrático médio) em comparação aos outros atributos de Share% e Rat%. Em relação a precisão (treinamento), Fid% foi precisamente melhorado, saindo de aproximadamente 0.63 e indo para 0.98.

4.6 Comparação de Modelos

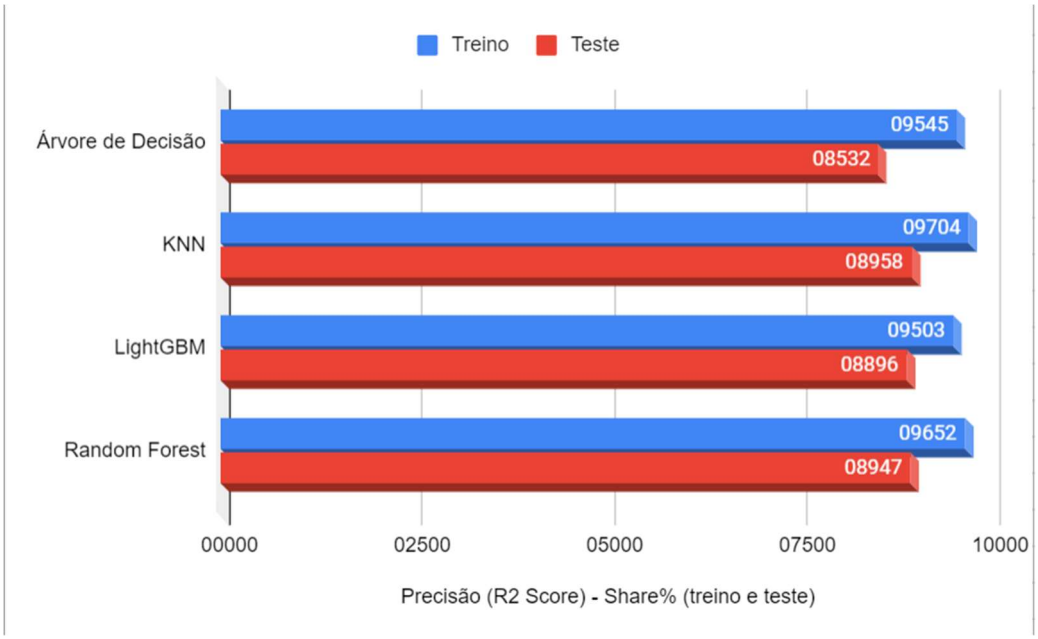
Por meio dos gráficos abaixo, foi realizada uma análise para determinação e comparação dos modelos que obtiveram os melhores resultados em termos das métricas R2 e MSE, selecionadas para avaliação dos modelos.

Comparação da precisão do Rat% entre os modelos com R2 Score



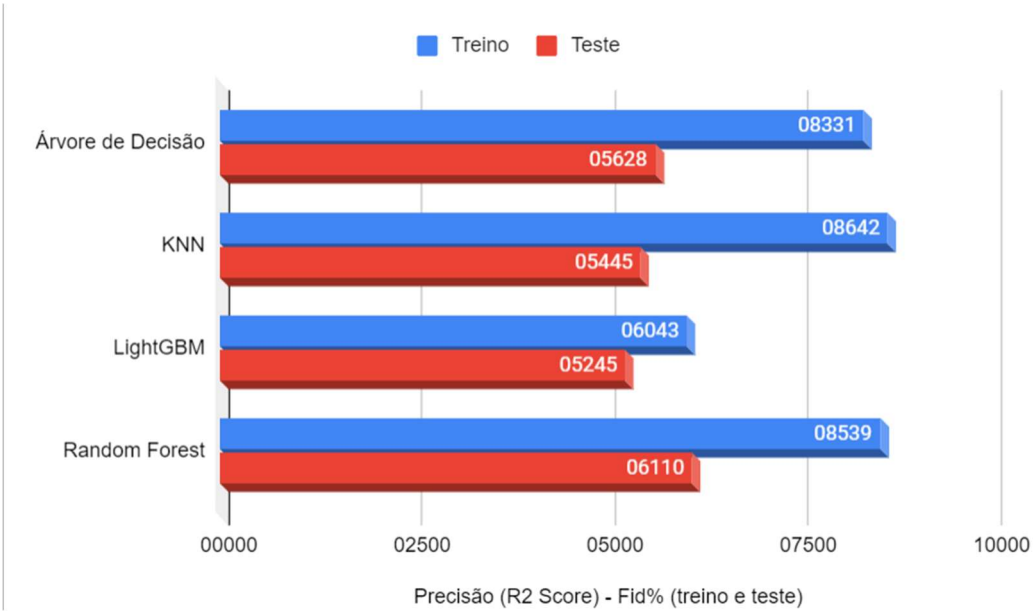
Para a coluna de Rat%, o modelo de árvore de decisão obteve a maior diferença entre os valores de treinamento e teste do modelo. O modelo KNN obteve o melhor valor de teste ficando bem próximo do valor obtido pelo LightGBM, isso demonstra que ambos os modelos estão com uma excelente precisão para a coluna Rat%.

Comparação da precisão do Share% entre os modelos com R2 Score



De modo geral, para a coluna Shr%, todos os modelos tiveram uma queda de aproximadamente 10%, mas os modelos KNN e Random Forest obtiveram os melhores resultados. Como são utilizadas todas as colunas de Shr% presentes no dataset para realização da predição, é possível inferir que uma manipulação nos inputs dos modelos selecionando apenas algumas colunas, poderia promover maiores resultados precisos. No entanto, o objetivo do modelo elaborado neste projeto, propõe uma predição realizada com todas as colunas presentes no modelo, sendo que estas posteriormente poderão ser manipuladas aos critérios estabelecidos para futuros projetos.

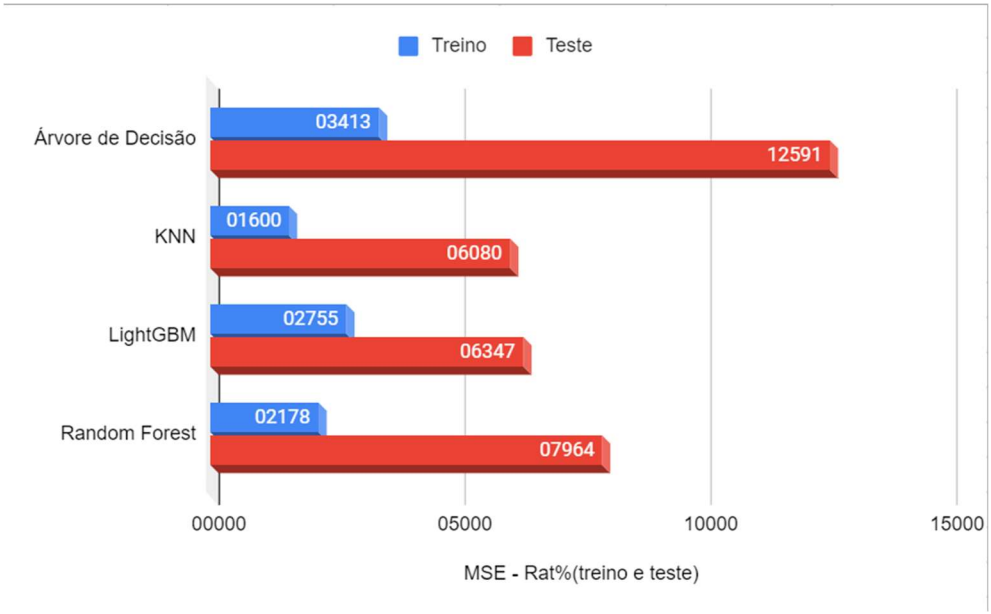
Comparação da precisão do Fid% entre os modelos com R2 Score



Os modelos apresentaram uma queda nos valores de precisão para a coluna Fid% comparando teste e treino. Todavia, o modelo LightGBM demonstrou estabilidade,

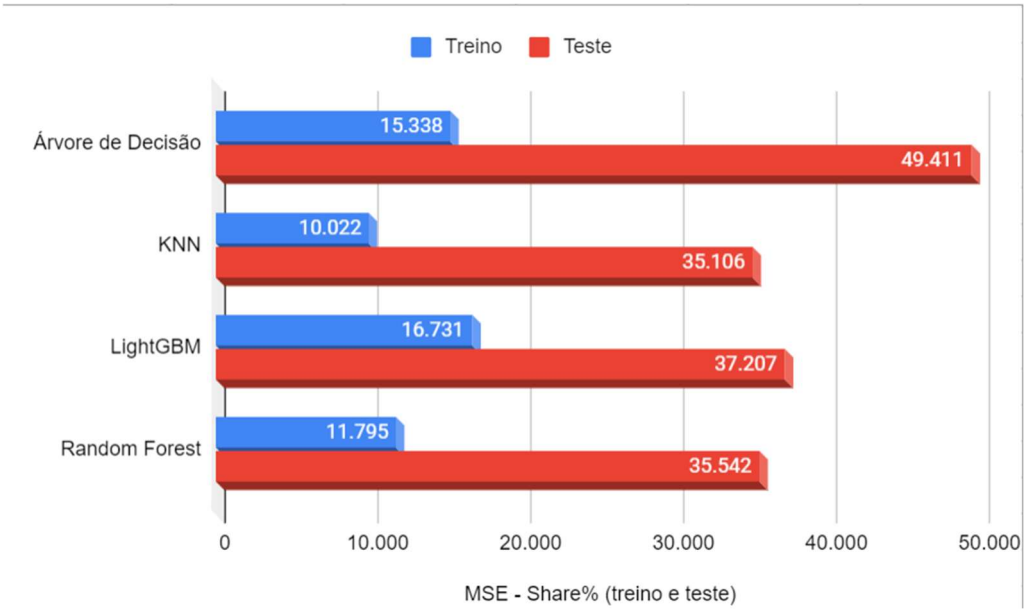
tendo uma diferença de apenas 8% nos valores, mas a baixa precisão de todos os modelos demonstra que a predição não está adequada para a tabela de Fid%, podendo inferir-se que o problema seja a quantidade de colunas utilizadas como output também para Fid%.

Comparação do MSE (Erro quadrático Médio) do Rat% entre os modelos



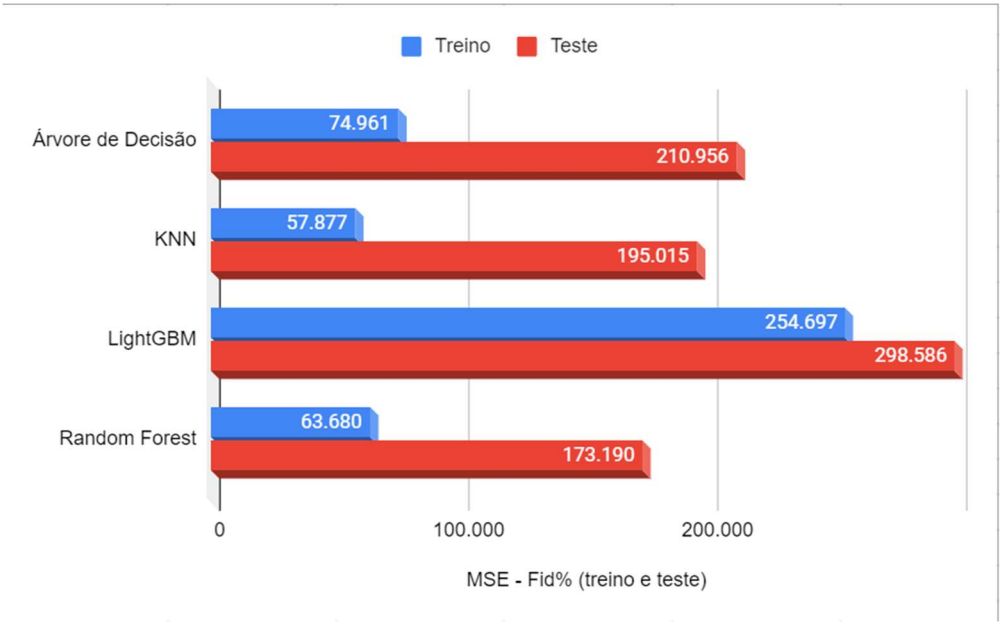
O MSE da coluna de Rat% para os modelos apresentou uma diferença excessiva entre teste e treino. Contudo, nos modelos mais adequados para esse projeto, o KNN e LightGBM, os valores não ultrapassaram zero, indicando que os modelos estão com uma acurácia de predição alta, indicando que os valores resultantes da predição estão próximos da linha de regressão,onde estão localizados os pontos resultantes da predição. O modelo de Árvore de decisão demonstra a ocorrência de overfitting, uma vez que a métrica MSE é sensível a erros e o valor de teste foi maior que 0.

Comparação do MSE (Erro quadrático Médio) do Share% entre os modelos



Para a coluna de Shr%, os valores de teste obtidos foram elevados para todos os modelos. A predição ainda é considerável por não se destoar tanto dos valores utilizados no treino, mas pode-se concluir que os valores estão divergentes do que se é considerado adequado levando-se em consideração que o MSE é uma métrica que quanto mais baixo o valor, mais próximo da realidade a predição estará.

Comparação do MSE (Erro quadrático Médio) do Fid% entre os modelos



Em conclusão, a coluna de Fid% obteve os piores valores de performance para todos os modelos. O modelo LighGBM obteve a maior taxa de erro, mas demonstrou certa estabilidade pelos valores de teste e treinamento estarem próximos. Em geral, os valores obtidos estarem altos em todos os modelos, pode indicar que os dados

existentes no dataset não são adequados para realização da predição dessa coluna ou/também deve-se ser levado em consideração maiores manipulações dos outputs considerados para a o Fid%.

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

BAHMANI, MJ. **Understanding LightGBM Parameters (and How to Tune Them)**. [S. l.], 21 jul. 2022. Disponível em: https://neptune.ai/blog/lightgbm-parameters-guide. Acesso em: 19 set. 2022.

CHIN, Mary. **Min_sum_hessian: a LightGBM demo.** [S. l.], 2020. Disponível em: https://www.kaggle.com/code/marychin/min-sum-hessian-a-lightgbm-demo/notebook. Acesso em: 19 set. 2022.

CUI, Jingsong; SEREDAY, Scott; , VP. **Using Machine Learning to Predict Future TV Ratings In An Evolving Media Landscape.** Nielsen, 2016

MA, Nan. **Prediction of Television Audience Rating Based on Fuzzy Cognitive Maps with Forward Stepwise Regression: International Journal of Pattern Recognition and Artificial Intelligence,** 2016.

BONDADE, Navi. **With AI Fox Film Studio Predicts Movie's Audience By Analyzing It's Trailer.** India, 2018.

HYAH, Suzya. **Hessian, second order derivatives, convexity, and saddle points.** [S. l.], 5 abr. 2018. Disponível em: https://suzyahyah.github.io/calculus/2018/04/05/Hessian-Second-Derivatives.html. Acesso em: 19 set. 2022.

Lightreading. **Gracenote launches 'Audience Predict' tool to gauge content performance.** Emeryville, Calif, 2021.

LIAO, Shannon. **BBC will use machine learning to cater to what audiences want to watch.** New York, 2016.

VAZÉ, Achyut. **Building a Model for Predicting TV Ratings.** Mumbai, 2016.

XIA, Jhiazi. **TVseer: A visual analytics system for television ratings.** Hangzhou, 2020.

AKULA, Ramiya, WIESELTHIER, Zachary, MARTIN, Laura, GARIBAY, Ivan. **Forecasting the Success of Television Series using Machine Learning**, Orlando, USA, 2019.

Istoé Dinheiro. **BRASIL empobrece em 10 anos e tem mais da metade dos domicílios nas classes D e E**, 2022.

HOAKE, Jake. **Machine Learning: Pruning Decision Trees**, Displayr.

ANAND, Akhil. **Post-Pruning and Pre-Pruning in Decision Tree**. Querala, 2020.

ARORA, Sarthak. **Cost Complexity Pruning in Decision Trees.** São Francisco, 2020.

RASTOGI, Rajeev & SHIM, Kyuseok. **PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning.** Korea, 2000.

T, Bex. **Kaggler's Guide to LightGBM Hyperparameter Tuning with Optuna in 2021.** [S. l.], 3 set. 2021. Disponível em: https://towardsdatascience.com/kagglers-guide-to-lightgbm-hyperparameter-tuning-with-optuna-in-2021-ed048d9838b5. Acesso em: 19 set. 2022.

ZHEREBTSOV, Danil. **Effortlessly tune LGBM with optuna.** [S. l.], 29 dez. 2021. Disponível em: https://danilzherebtsov.medium.com/effortlessly-tune-lgbm-with-optuna-49de040d0784. Acesso em: 19 set. 2022.

FRAJ, Ben, Mohtadi. **InDepth: Parameter tuning for Decision Tree.** Medium, 2017.

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Anexos

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