PredTv
Rede Gazeta

inteli
instituto
de tecnologia
e liderança

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
08/08/2022	Antonio Teixeira Sophia Tosar	1.1	Adição do Canvas de Proposta de Valor e Análise SWOT.
10/08/2022	Giovanna Rodrigues	1.2	Adição da Matriz de Risco.
11/08/2022	Giovanna Rodrigues	1.3	Adição das seções 1 e 2.
12/08/2022	Giovanna Rodrigues Sophia Tosar	1.4	Adição da seção 4.1
26/08/2022	Antonio Teixeira Daniel Barzilai Giovanna Rodrigues Sophia Tosar	1.5	Adição seção 4.2 e 4.3
08/09/2022	Antonio Teixeira Daniel Barzilai Sophia Tosar	1.6	Adição seção 4.4
11/09/2022	Antonio Teixeira Daniel Barzilai Emanuel Costa Sophia Tosar Thomas Barton	1.7	Adição seção 4.5
25/09/2022	Antonio Teixeira Daniel Barzilai Emanuel Costa Giovanna Rodrigues Sophia Tosar	1.8	Adição seção 4.6
05/10/2022	Giovanna Rodrigues Sophia Tosar	1.9	Revisão

Sumário

1. Introdução

A Rede Gazeta, fundada em 1928 com a primeira edição impressa do jornal A Gazeta, atualmente, é composta pelo site de notícias A Gazeta, pelas rádios CBN Vitória, Gazeta FM, Rede Litoral e Mix Vitória, pelas quatro emissoras da TV Gazeta (Grande Vitória, Norte, Noroeste e Sul) e pelos portais g1 ES e GE ES, e o maior grupo de comunicação multimídia do Espírito Santo. Afiliada à TV Globo, a emissora TV Gazeta atinge 41% da população capixaba e tem como objetivo contribuir com o desenvolvimento e fortalecimento de seu estado.

Acerca de sua programação, por ser uma afiliada, ela deve seguir a grade principal da TV Globo, mas tendo slots disponíveis para os programas de sua preferência. Nesse contexto, mesmo possuindo dados históricos de audiência, a TV Gazeta ainda não possui uma forma de prever se a aceitação da população aos novos programas esperada é superada, causando ansiedade e insegurança aos responsáveis pela programação.

2. Objetivos e Justificativa

2.1. Objetivos

O objetivo da TV Gazeta principal é o retorno dos seus investimentos em novos programas, logo, para que isso aconteça, é imprescindível que o novo programa tenha uma boa audiência e um bom score.

Nesse objetivo, eles devem entender quais características, como horário e gênero, mais influenciam na notoriedade de seus novos produtos, para assim, conseguir criar novos programas que vão a favor de todas essas características, tendo maior certeza de sucesso.

2.2. Justificativa

Neste projeto será descrito o PredTv, uma plataforma baseada em *machine learning capaz* de prever pontuações de audiência de programas futuros de acordo com as características escolhidas pela produtora, além de conseguir identificar quais dessas características têm mais peso sobre o score fornecido. Assim, a produtora consegue ter mais claridade sobre os elos fracos de seu novo produto e fortalecê-los com antecedência.

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

A metodologia CRISP-DM é uma sigla em inglês Cross Industry Standard Process for Data Mining, que em português significa Processo Padrão Inter-Indústrias para Mineração de Dados. Essa metodologia visa desenvolver modelos a partir da análise de informações e dados de um negócio para prever futuras falhas e soluções. A metodologia é dividida em seis etapas:

1- Entendimento dos Negócios: nessa etapa é definido o objetivo do projeto e as necessidades da empresa

2- Entendimentos dos Dados: É realizada a coleta de dados

3- Preparação dos Dados: Organização dos dados, essa etapa é a mais demorada da metodologia.

4- Modelagem: o tipo de modelagem a ser definida de acordo com a necessidade do negócio. Com a definição de qual modelo será utilizado, devem ser definidos quais atributos serão variáveis na construção desse modelo

5- Avaliação: avaliar se o resultado corresponde à expectativa do projeto.

6- Implementação (deployment): o modelo é colocado em produção, de modo a agregar valor para o negócio.

3.2. Ferramentas

Nome	O que é	Em que foi utilizado	Versão
Google Colaboratory	Software que utiliza máquinas virtuais no formato de notebook.	Utilizado para o pré-processamento, criação de gráficos e do modelo.	-
Excel	Programa utilizado na criação, edição e leitura de planilhas, entre outras funções, como gráficos e cálculos.	Utilizado para visualização do dataset	-
Visual Studio Code	Editor de código redefinido e otimizado para criar e depurar aplicativos modernos da Web e da nuvem.	Utilizado para o pré-processamento, criação de gráficos e do modelo.	1.71
Flask	É um framework escrito em Python	Foi utilizado para integrar back e front-end da interface	2.2.2

3.3. Principais técnicas empregadas

Durante a etapa de entendimento do negócio da metodologia CRISP-DM, o grupo o Value Proposition Canvas, a Matriz de Riscos e a Jornada do Usuário. Todas essas ferramentas foram implementadas a fim de entender um pouco mais sobre o mercado que a Rede Gazeta está inserida e como nosso produto vai impactar o desempenho da emissora.

Além disso, durante a etapa de modelagem, foram utilizados os seguintes modelos: Regressão Linear, KNN, Random Forest, LGBM e Support Vector Machine, na seção 4.5 serão expostos os resultados de cada um desses modelos.

4. Desenvolvimento e Resultados

4.1. Compreensão do Problema

4.1.1. Contexto da indústria

Principais players

O público-alvo deste projeto é o setor de inovação da Rede Gazeta, que utilizará o modelo preditivo criado para antever a audiência de produtos criados de acordo com seu horário de exibição, data, idade, gênero e classe social do público-alvo. Uma vez que a previsão será direcionada à audiência da Rede Gazeta, serão considerados para análise os telespectadores da TV Gazeta. Tal público pode escolher livremente assistir qualquer canal de TV disponível em seu domicílio. Portanto, para esta análise de mercado, foram consideradas as três maiores emissoras de TV aberta além da Rede Globo, visto que A Gazeta é sua afiliada.

De acordo com o levantamento TV PNT TOP publicado pela Kantar IBOPE Media, as maiores emissoras de TV do Brasil em audiência no ano de 2021 foram: Globo: 10,11, Record: 4,27, SBT: 3,39 e TV Band: 0,96. A receita líquida destas mesmas emissoras no ano de 2021 foi de: Globo Comunicação e Participações S.A: R$14,4 bi, Sistema Brasileiro de Televisão: 1,4 bi, Record TV Rádio e Televisão Record S/A: R$ 130,503 mi (lucro líquido).

Modelo de Negócios

Merchandising – Mechan é o momento em que o apresentador para o programa para apresentar o produto. Assim, esta apresentação é paga pela empresa que publica o produto. Scripts com as marcas dos patrocinadores e textos sobre o produto fazem com que o merchandising também gera, naturalmente, renda para a emissora.

Espaço Publicitário – Os espaços publicitários são momentos de pausa na programação para os horários comerciais. Esses comerciais existem desde 1951, eles entraram um ano após a chegada da televisão no Brasil.

Aluguel de Horário – Aluguel de horário são momentos em que programas de televendas ou programas de igrejas alugam um tempo da programação. Dessa forma, alugar um horário é uma forma de obter uma receita fixa para a emissora.

Tendências

Mesmo com o crescimento do uso de serviços de *streaming*, 79% da população brasileira absorve conteúdos audiovisuais por meio de TV aberta e canais por assinatura.

5 Forças de Porter

Ameaça de produtos substitutos:

Atualmente, vivemos na era da tecnologia, logo, serviços de streaming se tornaram cada vez mais populares, pois oferecem uma variedade de conteúdos e são mais práticos, uma vez que os usuários têm acesso por outros dispositivos, como computador e celular. Contudo, não se espera que isso se torne uma grande ameaça, com a criação da sua própria plataforma de streaming, a Globoplay, a Globo se inseriu nesse mundo da tecnologia e têm sucedido. No último trimestre deste ano, a arrecadação financeira pela Globoplay aumentou em 50%, animando a rede televisiva. Além do fato das televisões persistirem em quase todas as casas brasileiras e terem seu consumo impulsionado com a pandemia, portanto, se torna improvável a migração de usuários para outros produtos além do espectro da Rede Globo.

Ameaça de entrada de novos concorrentes:

Entrar no ramo de televisão pode ser muito intenso, isso se deve aos altos custos de investimento, mas também, deve-se considerar a força das marcas já estabelecidas, uma vez que estão nesse mercado por muitos anos e já possuem o respeito e fidelidade da população, possuindo quase um monopólio na tv aberta.

Poder de negociação dos clientes:

Devido a grande polarização política atual, se observa uma grande movimentação contra as emissoras contrárias à ideologia de seus usuários e com isso, eles migram para emissoras concorrentes. Se observa que nos primeiros quatro meses deste ano, o Jornal Nacional possuiu o menor número de audiência de sua história, jornal que é frequentemente afrontado por questões ideológicas.

Poder de negociação dos fornecedores:

Os fornecedores dessa indústria são aqueles que produzem os programas televisionados. Nesse âmbito, possuímos duas categorias, os fornecedores nacionais e internacionais, entre esses, o poder de negociação é contrário. Entre os fornecedores nacionais, esse poder é pequeno, uma vez que a possibilidade de terem seus produtos transmitidos por emissoras de grande porte é diminuta. Enquanto os fornecedores internacionais, possuem a chance de distribuir seus serviços para um maior número de consumidores, como plataformas de streaming e diversas emissoras, possuindo um poder maior.

Rivalidade entre os concorrentes:

As grandes concorrentes estão estabilizadas no mercado por muitos anos, possuindo um monopólio de audiência, contudo, nota-se que mesmo com a migração de espectadores de uma emissora para outra, a Rede Globo continua possuindo o primeiro lugar entre elas. Um exemplo, é o fato mencionado anteriormente, apesar da grande perda de audiência pelo programa Jornal Nacional, ele continua sendo o maior telejornal brasileiro. A rivalidade é grande, já que o câmbio entre artistas é dificultado ou barrado, mas é inegável que o poder da emissora Globo é superior ao das outras emissoras.

4.1.2. Análise SWOT

Ambiente Interno	**FORÇA**	**FRAQUEZA**
	1.A criação do modelo preditivo pode auxiliar na medição de audiência. 2. A Rede Gazeta atende 41% da população capixaba. 3. Grande grupo de comunicação e multimídia do Espírito Santo.	1.O People Meter pode gerar dados enviesados. 2. Possuem um grupo pequeno na área de inovação 3. Prioriza o estado do Espírito Santo, que é um estado pouco populoso
Ambiente Externo	**OPORTUNIDADE**	**AMEAÇA**
	1. Aumento da audiência da Rede Gazeta com a criação do modelo preditivo. 2. Fortalecimento do desenvolvimento do Espírito Santo.	1. Competitividade de mercado 2. Aumento do uso de smartphones que afeta o tempo dos telespectadores conectados na TV.

4.1.3. Planejamento Geral da Solução

Ao lançar novos conteúdos, a Rede Gazeta deve decidir diversas variáveis que influenciam na audiência de seu produto, apesar de possuírem uma quantidade elevada de dados históricos de audiência, não é possível ter extrema certeza sobre um score futuro, logo, há muita insegurança no investimento de lançamentos, uma vez que há o medo de que não haja retorno financeiro.

Dentre esses dados históricos, tivemos acesso aos últimos três anos de audiência da própria TV Gazeta divididas em métricas importantes na área televisiva (mais contextualizadas na área 4.3 do documento) e entre diversos perfis sociais, como idade, gênero e classe socioeconômica. Assim como a de outras 4 concorrentes para comparação. Todos os dados foram fornecidos pela Rede Gazeta e devem permanecer confidenciais.

A partir disso, desenvolvemos o PredTv, um modelo preditivo que antecipa a audiência de acordo com variáveis adicionadas pelos usuários. Nosso produto vai ajudá-los a saber se um programa, que ainda não foi lançado, vai atender as expectativas esperadas. Além de informar o peso que cada variável tem sobre o score final, assim, a TV Gazeta tem maior controle sobre quais características do novo produto devem ser modificadas para uma maior aceitação.

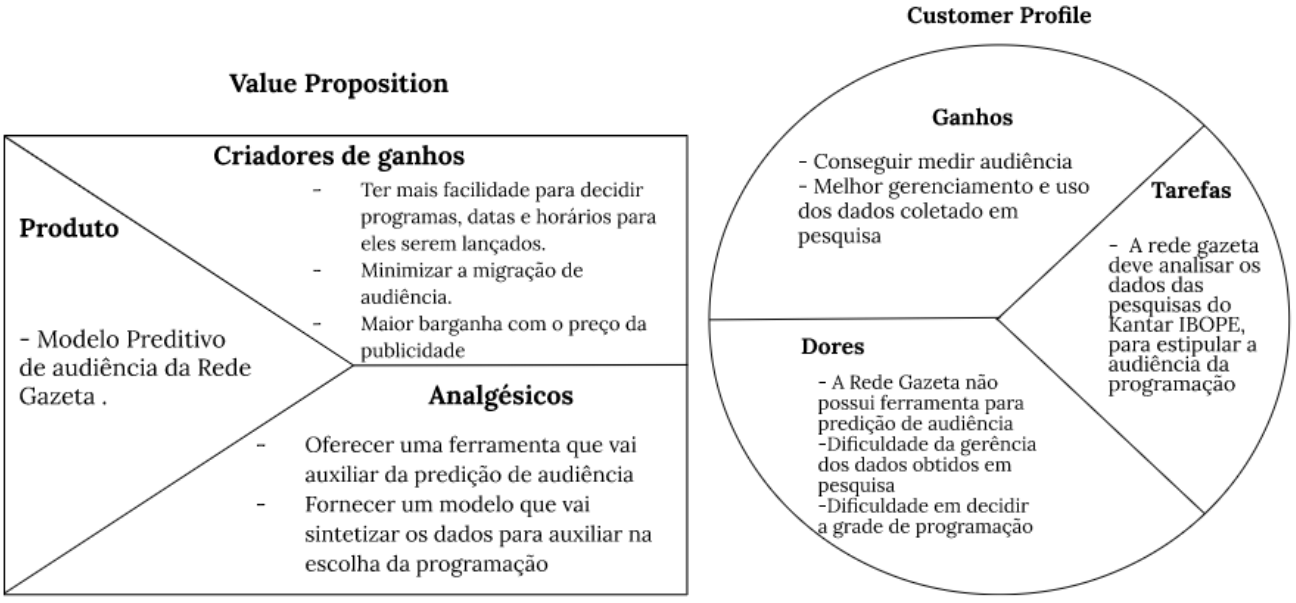
O tipo de tarefa utilizado será regressão, pois nós devemos prever uma resposta, estimando um valor numérico.

Posteriormente, o PredTv deverá ser utilizado pela equipe de marketing e programação da Rede Gazeta, de maneira que eles irão imputar as variáveis desejadas (horário, data, público-alvo) e o programa gerará um score baseado nesses inputs. Ao receber o resultado, a equipe deve ponderar se ele é o esperado e, se não, mudar as variáveis que mais influenciaram, elas serão informadas pelo próprio programa, até que o resultado seja satisfatório.

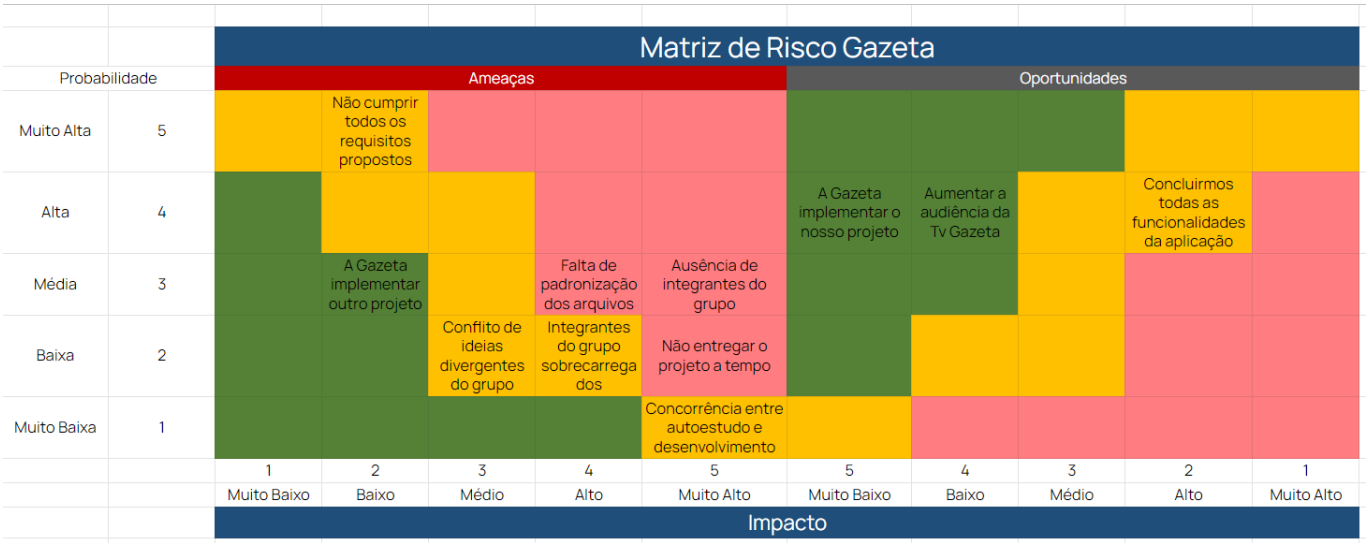
Consequentemente, a emissora terá maior facilidade na decisão das variáveis de um novo programa; terá um melhor gerenciamento e uso dos vários anos de pesquisa de audiência, podendo até diminuir esse armazenamento no futuro, uma vez que o programa já saberá o impacto das variáveis no seu resultado a partir dos anos usados anteriormente; conseguirá diminuir a migração de telespectadores; além de ter maior poder de barganha nos preços de publicidade e maior retorno financeiro.

Ao final, a solução será considerada um sucesso quando sua margem de erro for pequena ou desconsiderável.

4.1.4. Value Proposition Canvas



4.1.5. Matriz de Riscos



4.1.6. Personas

Luis Cassius Gomes, 36 anos, Diretor de inovações.

Biografia: Formado em tecnologia na área de análise e desenvolvimento de dados; fez pós-graduação em gestão corporativa; trabalha numa rede de TV; está há mais de 12 anos no ramo.

Características: Costuma buscar soluções através de tecnologia; gosta de estar envolvido com processos de inovação; busca soluções em automatização de processos; procura sempre manter-se atualizado; tem boa conexão com os coordenadores e diretores de TV.

Motivações com a plataforma: Melhorar a audiência dos programas da empresa; melhorar a assertividade na escolha da programação.

Motivação com o problema: Não possui um sistema customizado e flexível de predição para os novos produtos de sua programação..

Dores: Possibilidade de não estar aproveitando 100% o potencial de um programa; não possui uma aplicação auxiliar ao PeopleMeter; falta de métricas mais acuradas; ausência de um sistema no qual ele possa manipular os dados à vontade.

Santos da Silva, 60 anos, Apresentador e Produtor-chefe.

Biografia: Formado em Produção audiovisual; ganhou diversos concursos de curta-metragens antes de se formar; começou por baixo na empresa, sendo promovido até seu cargo atual; apresenta um programa na rede de maior audiência do estado.

Características: Apaixonado por arte audiovisual; personalidade cômica; domina a arte da liderança; possui entendimento amplo sobre a produção e desenvolvimento de programas de comédia.

Motivações com a plataforma: Saber quais programas priorizar (os que possuem maior audiência); aumentar a audiência de seu próprio programa.

Motivação com o problema: Não sabe o que deve ser feito para melhorar a audiência.

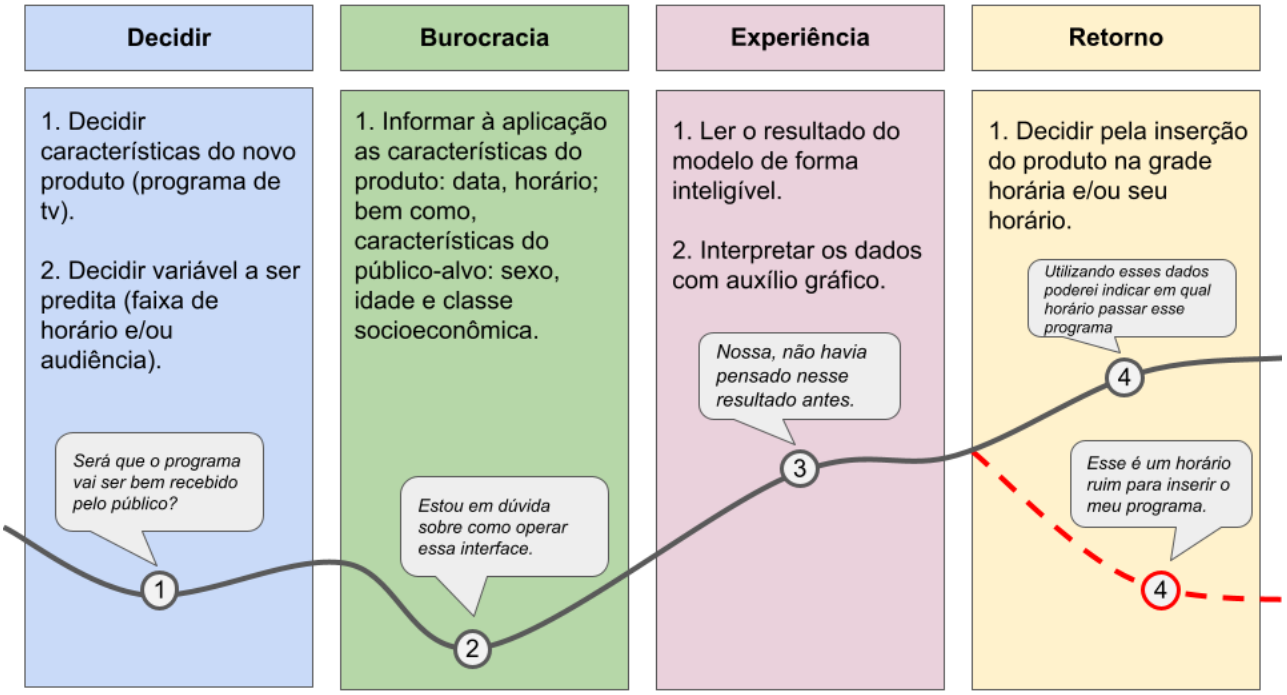
Dores: Não possui um sistema para medir o melhor programa com base nas audiências e público-alvo; carência de um modelo com informações de fácil leitura e compreensão.

4.1.7. Jornadas do Usuário

Luis Cassius Gomes

Cenário: Diretor de inovações testa a plataforma de predição para a predição do *score* de audiência de um novo programa sendo considerado para integrar a grade horária.

Expectativas: Diretor de inovações recebe, sem dificuldades, como resposta, a predição de audiência de acordo com as características informadas. No mesmo resultado também está presente o nível de influência que cada tipo de dado teve no resultado final.



O mapeamento demonstrado acima indica que temos como oportunidade a criação de uma interface para facilitar o uso do modelo criado. Desta forma, a experiência do usuário durante a realização das tarefas será mais intuitiva e agradável. No que diz respeito ao modelo, um efeito secundário da possibilidade de prever a audiência de forma qualitativa em faixas de horário específicas é a escolha de horários de publicidade. Por fim, uma vez que a plataforma for implementada em seu cenário ideal de utilização, esta poderá avaliar se suas predições estão de acordo com o comportamento da audiência no domínio empírico. Sendo assim, identifica-se a oportunidade de usar o confronto das predições com os dados empíricos recém-coletados para retroalimentar o modelo, tornando-o mais acurado.

4.2. Compreensão dos Dados

Dados fornecidos pela Rede Gazeta sobre seu histórico de audiência e grade de programação e a de seus concorrentes principais. Os dados informam valores de métricas importantes no mundo da televisão, tais como:

- rat (televisores conectados nessa emissora dentre os televisores possíveis)
- share (televisores conectados nessa emissora dentre os televisores ligados)
- fid (televisores que permaneceram conectados nessa emissora por mais de um minuto).

Eles foram medidos de 5 em 5 minutos por dois anos e categorizados em classes econômicas (AB, C1, C2, C3, DE), gêneros (feminino e masculino) e idades. Foram 438 MB de dados em formato XLSX.

A partir disso, podemos agregar os horários de audiência com a programação, dados que também foram enviados pelo cliente em uma nova planilha de formato XLSX e 5 MB de tamanho , que possui o nome dos programas juntos de sua categorização, monitorados de 5 em 5 minutos no período de 2 anos, e assim definir a relação entre telespectadores, horário e tipo de conteúdo.

A pesquisa é feita pela empresa Kantar IBOPE Media, de renome na área de pesquisas de audiência. Os dados são coletados através de dispositivos chamados PeopleMeter, que coletam o perfil do espectador e a emissora assistida. São, em média, 200 aparelhos distribuídos por residências com diversos perfis e que rotacionam a cada 2 anos para evitar algum viés na projeção.

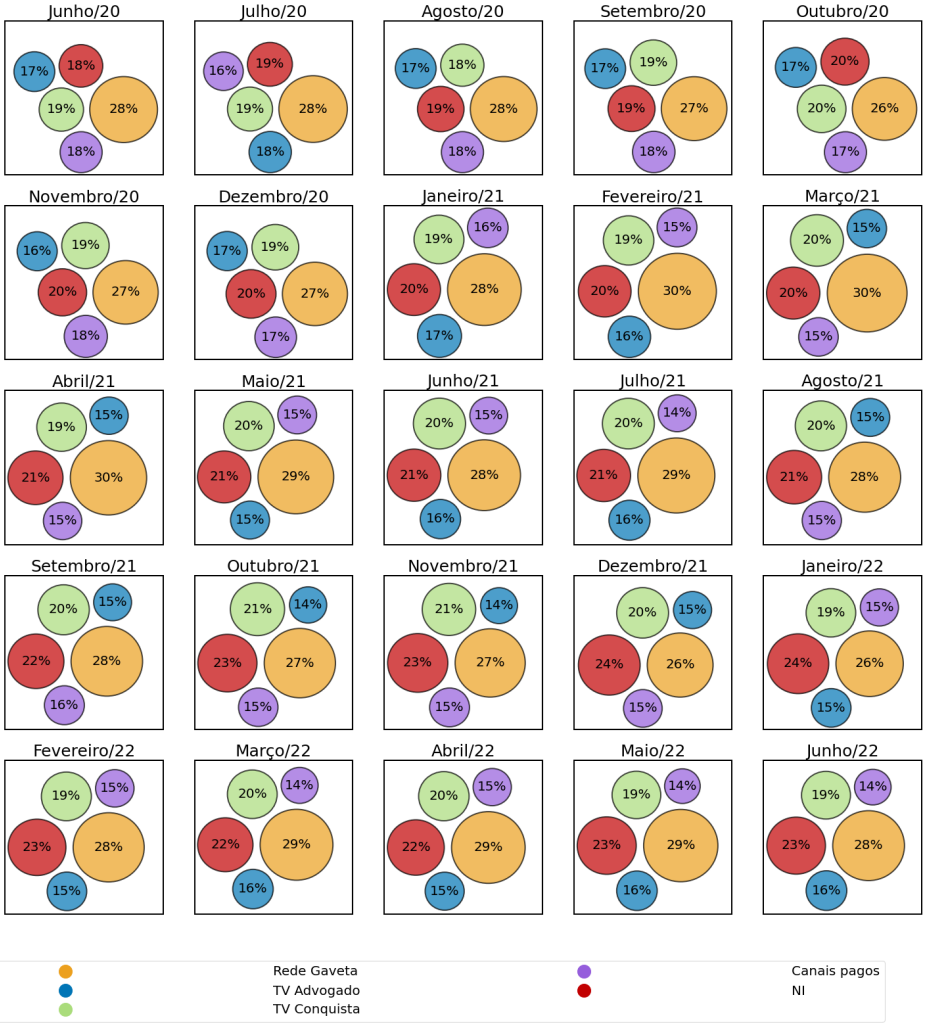
Posteriormente, fizemos subconjuntos para uma análise mais profunda, sendo esse:

- Planilha composta apenas pelo rat voltado para domicílios de todas as emissoras, a métrica de maior importância na análise de audiência. Podemos então, determinar horários de pico, médias de audiência de um nível geral.
- Planilha composta apenas pelo rat voltado para os gêneros, podendo então, determinar as médias de audiência por gênero e seus horários de pico.

Os dados são confidenciais, logo, não serão expostas os nomes das emissoras nesse documento.

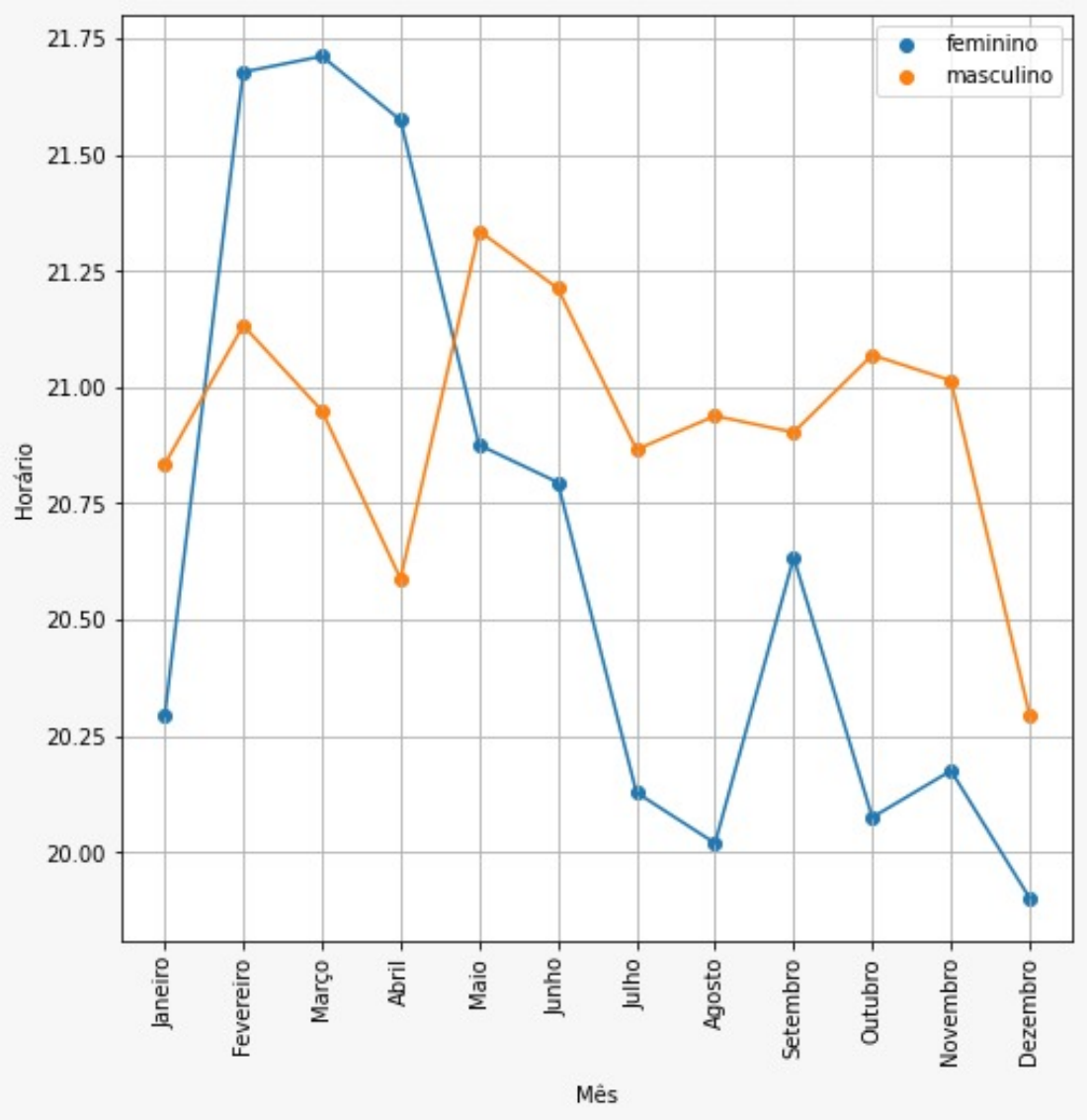
O resultado (*target*) dessa predição será o score de audiência, por ser um número aproximado da realidade e não um resultado binário entre “audiência boa” ou “audiência ruim”, sua natureza é contínua.

Gráfico de audiência das emissoras do Espírito Santo



O gráfico acima nos informa a média mensal nos anos de 2020 a 2022, é possível a partir daí, determinar que a Rede Gaveta possui uma predominância na maior audiência, sendo, em todos os meses, a maior audiência. Ademais, também é possível afirmar que as emissoras NI e Rede Conquista disputavam pela segunda colocação por muitos meses, com audiências iguais ou com diferença de um porcento, contudo, a partir do mês de setembro de 2021, a emissora NI possuiu uma diferença de dois porcento que vem aumentando desde então, demonstrando um fator de tendência nos últimos meses.

Gráfico de diferença de audiência por gênero



No gráfico acima, os dados demonstrados são resultados do horário com a maior média de audiência de todos os meses de janeiro, fevereiro e assim por diante. Inicialmente, achamos interessante calcular a média de audiência por horário em meses iguais, uma vez que supomos que esses possuiriam programações similares, contudo, esse gráfico está enviesado, já que meses iguais podem possuir programas cíclicos, logo, um programa passado no primeiro mês de janeiro desses dados, não estará mais passando no próximo mês de janeiro e será voltado para um público-alvo distinto, então, os horários de pico entre meses iguais são discrepantes e uma média entre eles não traria valores reais. Portanto, um novo gráfico será feito com as médias de horário de pico de todos os meses, trazendo mais sustentação para nossas hipóteses.

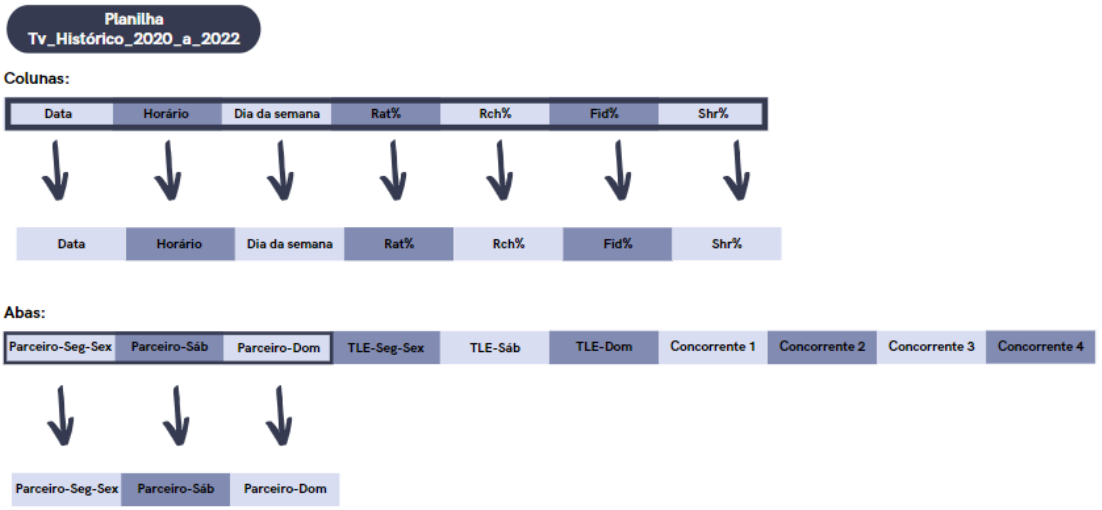
4.3. Preparação dos Dados

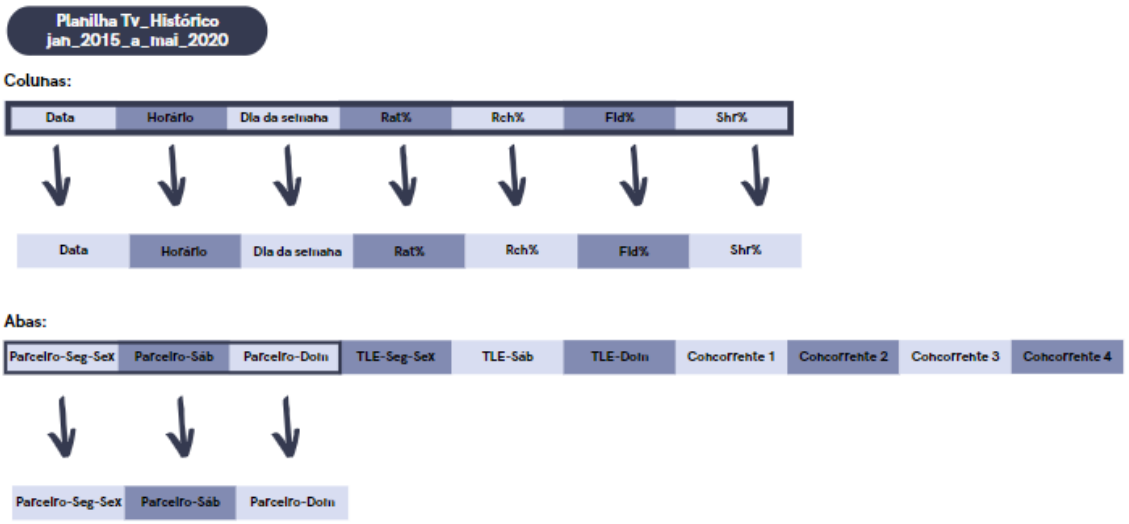
4.3.1. Otimização dos dados

Os dados iniciais possuíam um conteúdo muito grande, logo, o seu carregamento demandava muito tempo. Então, para a preparação dos dados foi necessário, inicialmente, a realização de uma nova planilha, em formato CSV, para que a manipulação dos dados disponibilizados pelo parceiro fosse feita de forma mais dinâmica e com o objetivo de obter um artefato que contivesse apenas os dados que iremos, efetivamente, utilizar na criação do modelo preditivo, tendo seu processamento agilizado.

O novo arquivo gerado foi feito com base em três abas, pertinentes ao parceiro, da planilha original "TV_Histórico.xlsx", uma com todos os dados referentes apenas aos dias úteis, enquanto as outras duas eram referentes ao sábado e domingo. Logo após, as colunas selecionadas para o novo arquivo foram aquelas referentes ao "rat%", métrica preferencialmente utilizada na medição de audiência. Com isso, os testes realizados nos modelos gerados, deixaram claro que shr% e fid% são necessários para maior acurácia do modelo, servindo como output da audiência prevista e, este que por sua vez gera as outras características que possuem rat como unidade de medida.

Após gerar esse CSV, foi disponibilizado pelo parceiro mais uma planilha com dados desde 2015, a qual foi refinada e mesclada com o csv já existente. Essa nova planilha foi submetida aos mesmos procedimentos da anterior, ordenação de dados, anonimização, exclusão das colunas não necessárias e mesclagem da mesma com o CSV definitivo.





4.3.2. Ordenação das datas

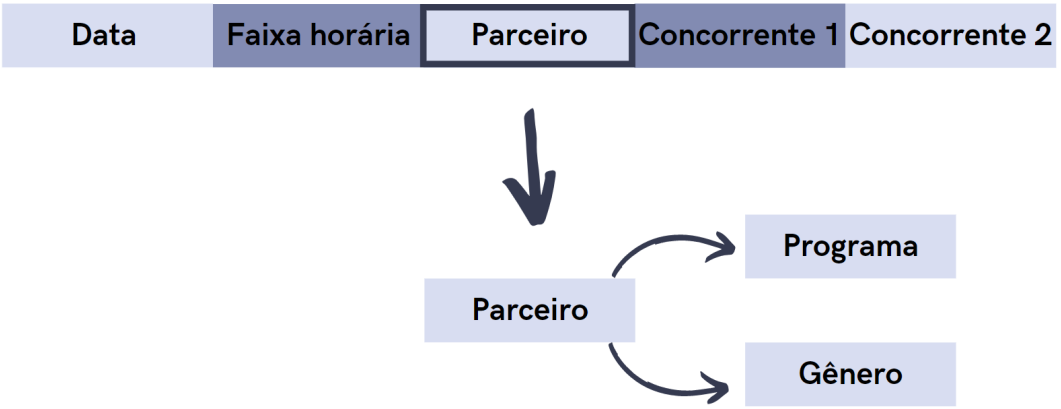
Ao adicionar as abas de final de semana no novo CSV, todos os sábados e domingos são posicionados no final da planilha, dificultando a visualização, uma vez que as semanas não possuem uma linearidade. Então, foi aplicada uma ordenação nas datas, movendo os sábados e domingos para suas devidas posições de acordo com o dia, mês e ano. Isso foi feito reorganizando a planilha por dia e depois por horário, necessariamente, nessa ordem, pois essas características são hierarquizadas. Após isso, utilizando uma lógica condicional e de substituição, as unidades de tempo foram limitadas, sendo limitadas como são na vida real, 24 horas por dia, por exemplo. Elas estavam representadas até 29 horas, o que dificultava a organização em dias, semanas, meses e consequentemente, anos.

4.3.3. Junção da grade horária

Até o momento, o arquivo principal continha apenas valores de audiência e espaços de tempo, mas não possuía a programação exibida. Então, através de outro documento xlsx “grade_Diária_06_2020_a_06_2022” fornecido pelo parceiro, foram movidos os dados a respeito do nome da programação e sua categoria para o novo CSV. Contudo, essas duas informações estavam numa única coluna, dificultando a visualização, então, foi necessário fazer a divisão destas em duas colunas: “Programa” e “Gênero”. Além disso, foi necessária a anonimização dos nomes dos programas, o que foi realizado utilizando lógica de programação e substituindo todos os nomes dos programas por “PROGRAMA“ acompanhado de um número.

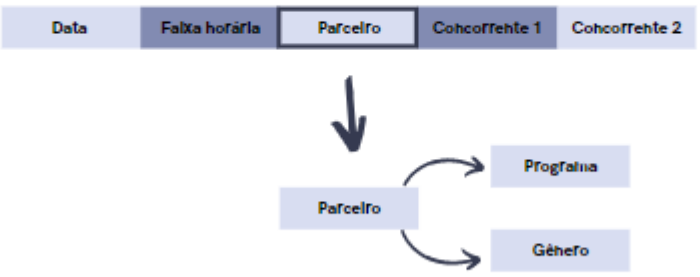
Planilha
grade_Diária_06_2020_a_06_2022.xlsx

Colunas:



Planilha
grade_Diária_jan_2015_a_mai_2020

Colunas:



4.3.4. Verificação de valores nulos ou ausentes

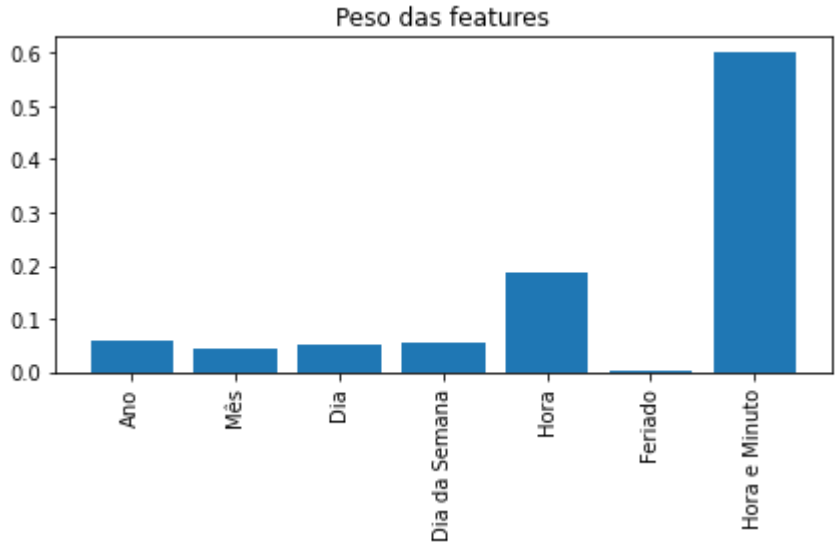
Foi feita uma verificação por valores nulos, mas, nenhum foi encontrado, logo, não foi necessário ser feito algum procedimento para removê-los ou substituí-los.

4.3.5. Seleção de Features

Para atingir o objetivo do PredTv, que é prever audiências de novos programas, é importante a análise do histórico, assim, é possível definir as features, características que influenciam diretamente no score final. Nas features que se referem a espaço de tempo, as mais importantes seriam data, dia da semana e horário de início do programa, uma vez que essas foram as que mais impactaram os modelos de predição. Além disso, também deixamos informações sobre o gênero do determinado programa passando no tempo em questão, e alguns Programas que se repetem durante os anos e que tem grande impacto, para determinados modelos, no total de audiência. Também, como foi visto que o minuto não tinha muito impacto como feature, mas, o minuto juntamente com a hora tinha, foi criado uma coluna representando a hora e o minuto em decimal, e ela se provou a feature com maior importância em quase todos os modelos. Os modelos de árvore de decisão se deram melhor com label encoding, enquanto, os lineares e o KNN se deram melhor com one hot encoding.

4.3.6. Peso das features

Utilizando o modelo Random Forest e seu atributo, já imbutido de "feature importance" podemos ver o quão impactante cada uma das features é para o modelo. Numa escala de 0 a 1, onde 1 é a soma de todos os graus de importância juntos, podemos ver no gráfico abaixo que, com uma liderança considerável, temos a coluna de hora e minuto dada como a mais importante seguida da informação de hora. As outras features temporais não são tão relevantes quanto estas, e possuem graus de importância próximos, mas, diferente delas, e em último lugar, temos a coluna de feriados.



4.4. Modelagem

Para a geração de modelos foram feitas algumas alterações nas planilhas e nos dados do CSV principal. Os outliers foram corrigidos criando mais duas colunas no nosso CSV principal, para realocar os outliers principais que seriam: audiência do reality show da emissora parceira e a audiência nos dias de feriados nacionais.

Ademais, para facilitar a comparação com os dados das outras emissoras foi realizado o processo de normalização para alguns modelos, que consiste em limitar os dados em um certo alcance gerando melhor proporcionalidade entre os dados das emissoras. Por fim, o único processo que foi realizado para a criação de todos os modelos utilizados foi a geração do CSV definitivo o que é descrito na seção 4.3 Separação de dados.

4.4.1. LGBM

Light GBM é um framework de algoritmo. Gradient Boosting é um método de Machine Learning para complicações de regressão e classificação, que apresenta um modelo preditivo, configurado em um conjunto de modelos de previsão fracos, principalmente as árvores de decisão. Como outros métodos de reforço, ele produz o modelo em partes, e os generaliza, possibilitando o aprimoramento de uma função de perda diferenciável arbitrária.

O principal objetivo do Light GBM é gerar uma cadeia de modelos fracos, no qual cada modelo tem como finalidade, reduzir ao máximo o erro do modelo prévio, mediante uma função de perda. Nos reparos de cada modelo fraco é multiplicado um valor, a taxa de aprendizagem. Tal valor é responsável por instituir o impacto de cada árvore no modelo final. Quanto menor o valor, menor a colaboração de cada árvore.

Portanto o LGBM foi utilizado, pois esse modelo é rápido na preparação, logo ele gera grande produtividade. Além disso, ele faz pouco uso de memória, pois substitui qualidades consistentes por receptáculos discretos.

4.4.2. KNN

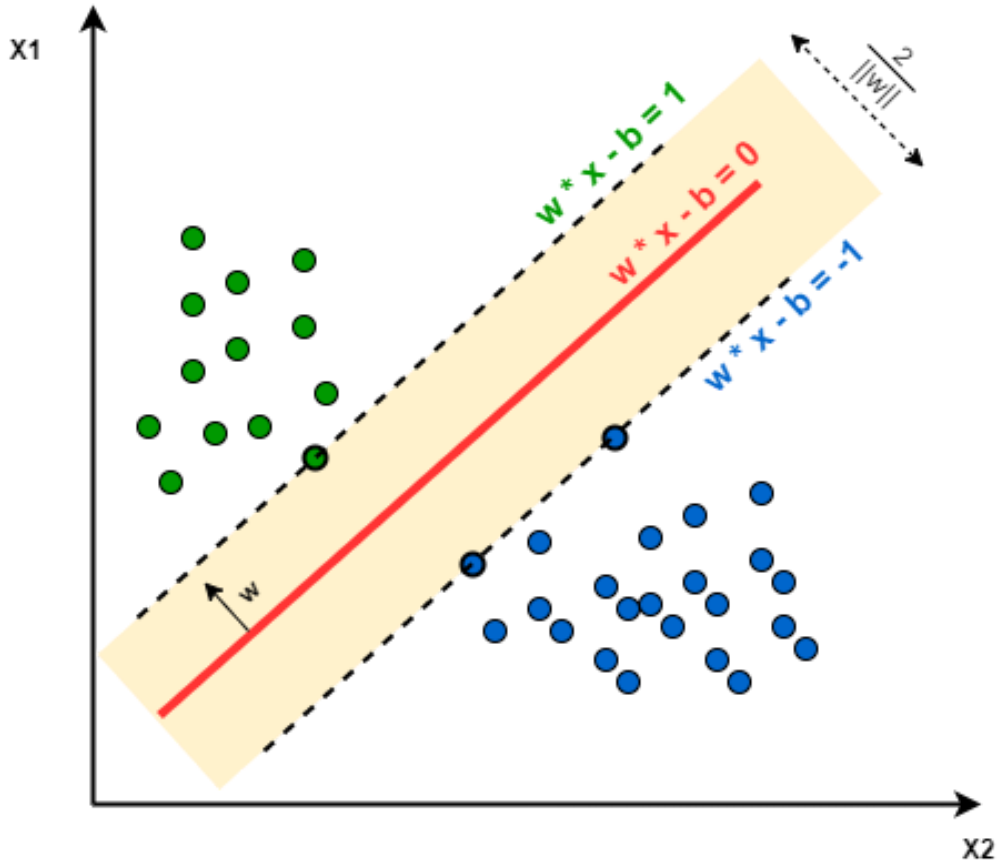
KNN, ou K-vizinhos mais próximos, consiste em um modelo de predição supervisionado baseado na distância com os seus vizinhos mais próximos. Primeiramente, um certo dado é escolhido, e com base em seus atributos, o KNN faz uma comparação com seus vizinhos mais próximos para definir o valor previsto. O k em questão representa o número de vizinhos que serão comparados com o dado escolhido que possui muita influência na comparação, tornando-a mais sensível quando baixo e menos sensível quando mais alto.

As vantagens que contribuíram na escolha do KNN como um dos modelos testados são: a simplicidade de implementação e a sua eficácia em diversas situações.

4.4.3. Support Vector Machine

Support Vector Machine é um algoritmo de aprendizado de máquina supervisionado que pode ser usado para desafios de classificação ou regressão. O objetivo desse algoritmo é criar a melhor linha ou limite de decisão que possa segregar o espaço n-dimensional em classes para que possamos facilmente colocar o novo ponto de dados na categoria correta no futuro. Esse limite de melhor decisão é chamado de hiperplano.

O SVM escolhe os pontos/vetores extremos que ajudam na criação do hiperplano. Esses casos são chamados de vetores de suporte. A imagem a seguir ilustra a forma como o SVM funciona:

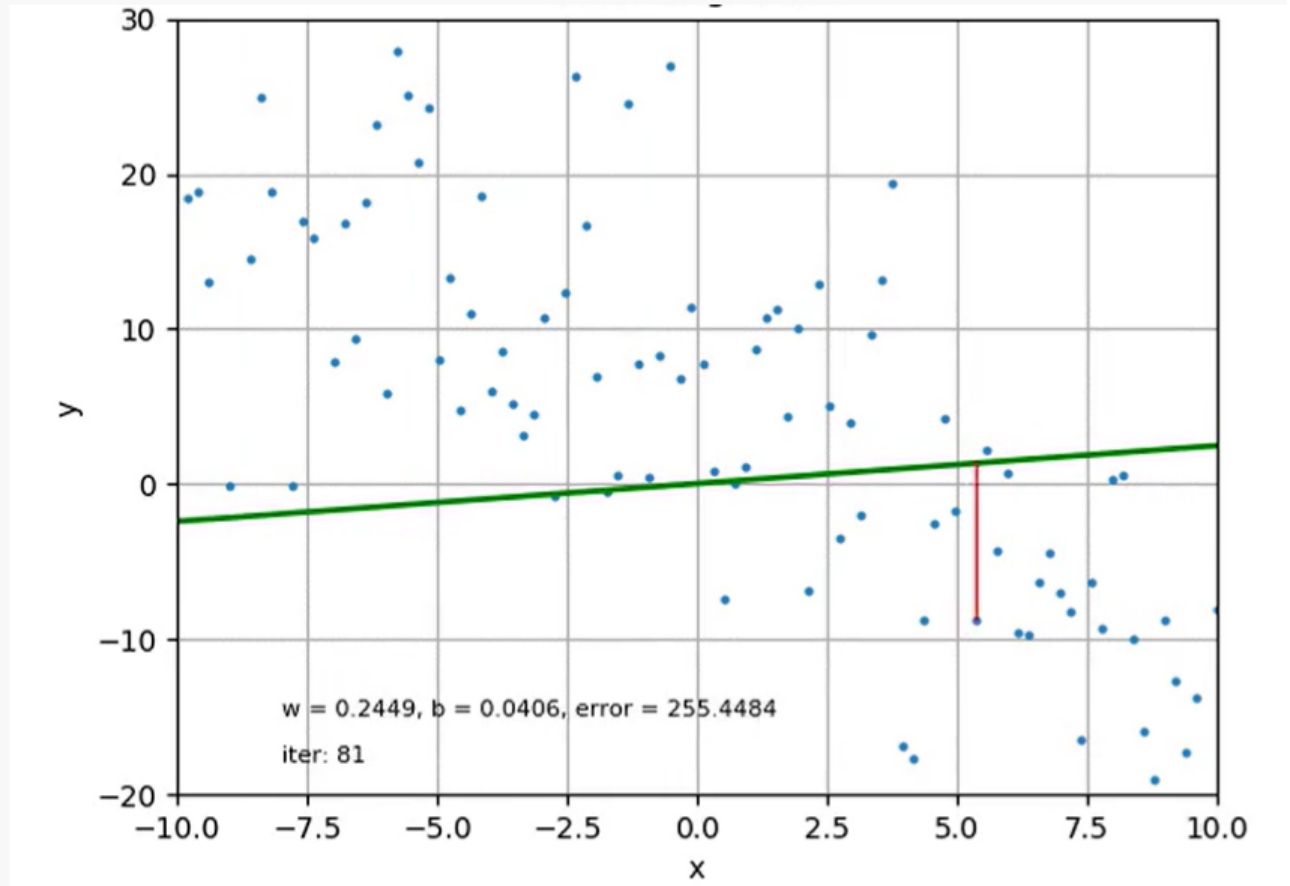


É importante ressaltar que o SVM foi utilizado, pois em caso de outlier, esse modelo busca a melhor forma possível de classificação e, se necessário, desconsiderar o outlier. Ademais, o SVM funciona bem em domínios complicados, em que existe uma clara margem de separação.

4.4.4. Linear Regression

Regressão linear consiste em um processo de traçar uma reta através dos dados disponibilizados em um diagrama de dispersão. A reta, quando bem acurada, pode gerar um resumo dos dados, permitindo a criação de um modelo supervisionado preditivo de dados. Esse tipo de modelo foi escolhido por se tratar de um algoritmo de predição mais usual e mais simples de ser feito, ele é bem útil quando não se possui muitos outliers e isso ocorre porque quando existem muitos pontos fora da curva, a reta fica refém dos pontos mais afastados.

Os dados foram submetidos ao processo de normalização padrão, que consiste em limitar os dados entre 0 e 1, e esse processo é útil, pois deixa os dados dimensionados da mesma forma o que resulta em uma melhor relação de proporcionalidade entre os valores das diferentes emissoras.



4.4.5. Random Forest

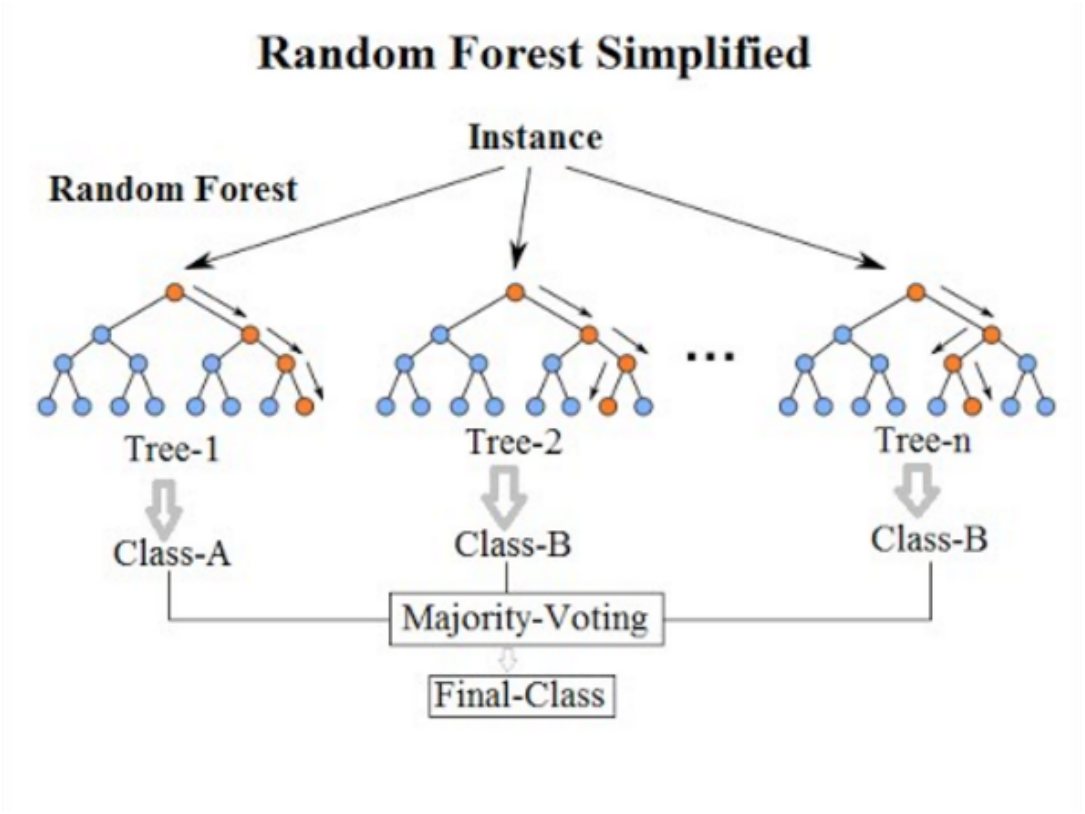
O Random Forest é um algoritmo que cria muitas árvores de decisão, de maneira aleatória. Cada árvore será utilizada na escolha do resultado final. As árvores de decisão estabelecem regras para a tomada de decisão. Dessa forma, o modelo criará uma estrutura similar a um fluxograma, com “nós” onde uma condição é verificada, e se atendida o fluxo segue por um ramo, caso contrário, segue por outro.

Esse modelo possui quatro etapas principais:

1. Seleção aleatória de algumas features
2. Seleção da feature mais adequada para a posição do nó raiz
3. Geração dos nós filhos
4. Repetir os passos acima até atingir a quantidade de árvores necessárias

Depois que o modelo é criado, as previsões são feitas a partir de "votações", cada árvore toma uma decisão com base nos dados apresentados, logo a decisão mais votada é a resposta do algoritmo.

O Random Forest foi utilizado, pois tem suas origens na forma mais básica e inicial de um algoritmo de suporte à decisão. Além disso, esse modelo resolve problemas de regressão e de classificação e costuma apresentar bons resultados. A seguir é mostrado, por meio de um esquema, como funciona o Random Forest:



4.4.6. Ajuste de hiperparâmetros

Hiperparâmetros são variáveis ajustáveis das quais o processo de aprendizado do modelo é dependente, uma vez que estes controlam a forma que o processo de aprendizado é realizado.

Contudo, nos diferentes modelos escolhidos anteriormente, os valores de hiperparâmetros possuem um valor padrão, no entanto, esses valores podem não ser os mais adequados para a finalidade da aplicação. Na tentativa de encontrar os valores que melhor

correspondessem ao nosso objetivo, inicialmente, esses foram mudados manualmente, cada vez que isso ocorria, o processo de treinamento era realizado novamente para a análise do impacto dessa mudança na acurácia final.

Todavia, a realização de testes manuais não possui uma boa eficácia, já que há a possibilidade de diversos valores para diversos hiperparâmetros, resultando em um valor elevado de combinações e um trabalho árduo na testagem de todas elas. Logo, foi necessário o uso de ferramentas que automatizassem esse processo: Grid Search, Random Search e Optuna.

- Grid Search: é uma técnica exaustiva, isso significa que ela testará todas as possíveis combinações, então pode levar um tempo para o seu processamento, contudo, te retornará o melhor resultado e os melhores hiperparâmetros entres os quais lhe foi dado.

- Random Search: essa técnica consiste na testagem de uma quantidade pré-definida pelo usuário, fazendo combinações aleatórias entre os hiperparâmetros passados e retornando o melhor resultado.

- Optuna: é um framework de otimização de hiperparâmetros utilizado particularmente para machine learning.

Os resultados da otimização estão descritos a seguir:

4.4.6.1. Support Vector Machine

Os hiperparâmetros mais críticos para o SVM são: Kernel, C e Gama. A função do kernel é transformar o conjunto de dados de treinamento em dimensões mais altas para transformá-lo linearmente separável. Além disso, “C” é o parâmetro de regularização l2, logo quando “C” é pequeno, a penalidade por erro de classificação é pequena. Ademais, o parâmetro gama pode impactar muito o desempenho do modelo, pois quando gama é pequeno, o raio de influência do vetor de suporte é grande. Os valores definidos anteriormente foram os padrões do modelo e, também, os resultados do Grid Search.

A seguir, estão os resultados depois da utilização do Grid Search:

```
The best accuracy score for the training dataset is 0.9693
The best hyperparameters are {'C': 1.0, 'gamma': 'scale', 'kernel': 'rbf'}
The accuracy score for the testing dataset is 0.9825
```

4.4.6.2. Random Forest

Os hiperparâmetros escolhidos para o modelo são: n_estimators, o número de árvores na floresta, e max_depth, tamanho máximo das árvores.

Foi utilizado o método Grid Search para escolha dos melhores parâmetros, a seguir, o resultado do processo:

```
Best Parameters:  {'n_estimators': 6, 'min_samples_split': 6, 'min_samples_leaf': 3, 'max_features': 'auto', 'max_depth': 10, 'bootstrap': True}
```

4.4.6.3. LGBM

Para o modelo LightGBM, foram utilizados os hiperparâmetros max_depth e num_leaves. Além desses, num_iterations e learning_rate, porém uma vez que quanto maior o num_iterations e menor o learning_rate forem, melhor a qualidade da predição, não se faz necessário a testagem. Como valores iniciais para num_leaves, foi utilizado o valor de 8, e para max_depth, 16. Então, após a utilização da biblioteca Optuna, utilizada para descobrir os melhores parâmetros, foi percebido que, quanto maior o max_depth e num_leaves, melhor a qualidade mas maior o tempo necessário para o aprendizado do modelo, também. Por isso, o grupo decidiu por utilizar um meio termo, no qual o valor não mudaria tanto, e não haveria necessidade de tanto tempo de execução. Os valores ficaram de 512 para max_depth, e 64 para num_leaves.

Parâmetros iniciais:

```
hyper_params = {
    'task': 'train', #função
    'boosting_type': 'gbdt', #tipo da regressão (gbdt, rf, dart, goss)
    'objective': 'regression', #tipo do modelo
    'metric': ['l1','l2'], #erro médio quadrático e erro médio absoluto
    'learning_rate': 0.25, #velocidade do aprendizado
    'feature_fraction': 1, #porcentagem do dataframe a ser utilizado
    'verbose': 0, #usado para retirar possíveis bugs existentes (debug)
    "max_depth": 16, #limita o tamanho de cada árvore
    "num_leaves": 8, #limita o número de folhas que cada árvore pode ter
    "num_iterations": 40_000, #quantidade de tentativas
}
```

Melhores parâmetros segundo Optuna:

```
FrozenTrial(number=110, values=[0.8737267180670986], datetime_start=datetime.datetime(2022, 9, 25, 21, 57, 40, 65007),
datetime_complete=datetime.datetime(2022, 9, 25, 21, 58, 5, 986412), params={'max_depth': 124, 'num_leaves': 511}, distributions={'max_depth':
IntDistribution(high=128, log=False, low=16, step=1), 'num_leaves': IntDistribution(high=512, log=False, low=64, step=1)}, user_attrs={},
system_attrs={}, intermediate_values={}, trial_id=110, state=TrialState.COMPLETE, value=None)
```

4.4.6.4. Regressão linear

O modelo de regressão linear foi o pior modelo testado pela equipe e não possui hiperparâmetros ajustáveis. Isso se deve ao fato de ser uma operação matemática que sempre utiliza as mesmas variáveis de entrada. O que pode ser feito para melhorar o modelo é limitado à seleção de features, uma vez que nessa etapa se pode, por exemplo, normalizar dados de diferentes formas para melhorar o processo de aprendizado do modelo.

4.4.6.5. KNN

Inicialmente, os hiperparâmetros escolhidos para o modelo foram n_neighbors e p. N_neighbors se refere ao número de vizinhos utilizado no modelo, o seu padrão é 5, mas pode ter um valor no intervalo de 2 até 100. O p define qual distância será utilizada quando nossa métrica é a chamada Minkowski, se é equivalente a 1, utilizamos a distância Manhattan (l1), mas se equivale a 2, a distância Euclidiana (l2) é a escolhida.

$$d_{euclidean} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad d_{manhattan} = \sum_{i=1}^{n}|x_i - y_i|$$

Ao utilizar o método de Grid Search, é claro que os melhores hiperparâmetros para nossa aplicação seria utilizat 2 vizinhos e a distância Manhattan. A seguir, a demonstração de como o Grid Search foi utilizado, trazendo uma acurácia final de 98%:

```
from sklearn.model_selection import GridSearchCV
```

```
grid_params = { 'n_neighbors' : [2,5,7,9,11,13,15,17,19,21,23,25,27,30],
                'p' : [1,2]}
```

```
gs = GridSearchCV(KNeighborsRegressor(),grid_params)
```

```
g_res = gs.fit(X_train, y_train)
```

```
print(g_res.best_score_)
```

```
0.9864825298375894
```

```
print(g_res.best_params_)
```

```
{'n_neighbors': 2, 'p': 1}
```

4.4.7. Modelo escolhido

Após decidir os melhores hiperparâmetros para cada modelo e realizar os testes que serão descritos nas seções 4.5 e 4.6, fica decidido que o melhor modelo para o objetivo da Rede Gazeta é o KNN.

4.5. Avaliação

As variáveis utilizadas para estudar os modelos de regressão foram: Coeficiente de determinação (r^2), que é uma medida de ajuste do modelo estatístico linear generalizado; o Nível de significância (ρ), que representa a probabilidade de rejeição da hipótese nula quando ela é verdadeira; o Desvio padrão(σ)que é uma medida que expressa o grau de dispersão de um conjunto de dados; Variância (σ^2) que mostra a medida da distância de cada valor do conjunto até o valor médio; e o Interquartil (box plot) que faz uma análise do grau de dispersão ao redor da medida da centralidade dos dados.

Além disso, outro ponto importante a ser ressaltado é a diferença entre os Dados de Treino e os Dados de Teste. Os dados de treino são aqueles apresentados ao algoritmo de Machine Learning para a criação do modelo. Por outro lado, os dados de teste são aqueles apresentados ao modelo após sua criação, simulando previsões reais e permitindo que o resultado final seja analisado

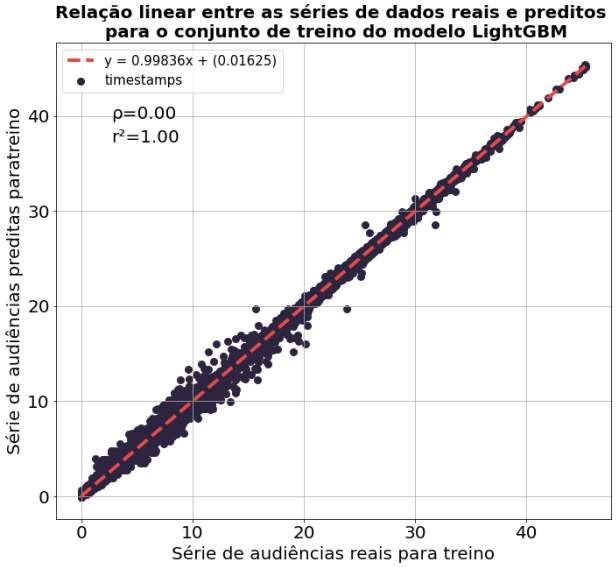
Ademais, deve-se destacar que em todos os modelos que o grupo testou, foi utilizada a regressão linear. A regressão linear é o processo de traçar uma reta através dos dados em um diagrama de dispersão, ela é de extrema utilidade para a realização de previsões .

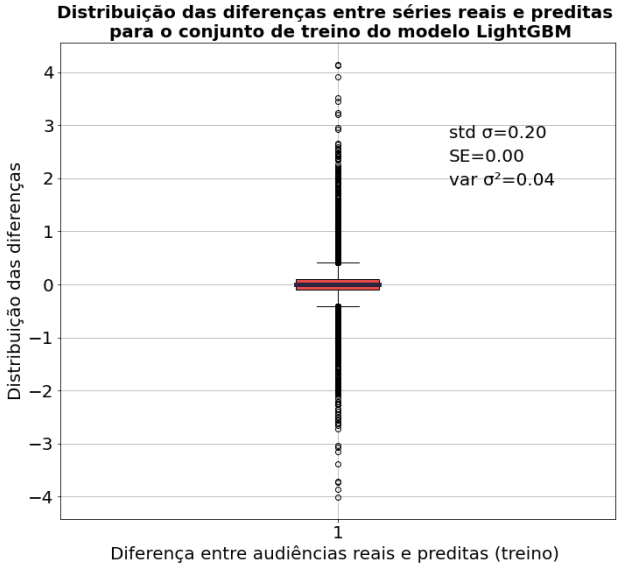
4.5.1. Resultados LGBM

Resultados Treino

Os resultados dos gráficos abaixo mostram que o treino do modelo LGBM apresenta um coeficiente de determinação igual a 1, uma variância de 0,04, um desvio padrão de 0,20 e erro médio quadrático de 0,03. Contudo, o valor do erro médio padrão não consta no gráfico, mas possui o resultado de 0,13. Além disso, não possui nível de significância.

Na primeira imagem, é notória a concentração dos dados reais e preditos na reta da equação linear, configurando uma boa acurácia do modelo. Já na segunda, concluímos que a metade das diferenças entre dados reais e preditos está muito próxima de zero, logo, não há um erro grande na predição de metade dos dados reais, além disso seus outliers estão, em sua maioria, localizados estre a faixa entre, aproximadamente, 0,5 e 2,5, tanto positivos quanto negativos, sendo assim, um resultado muito favorável.

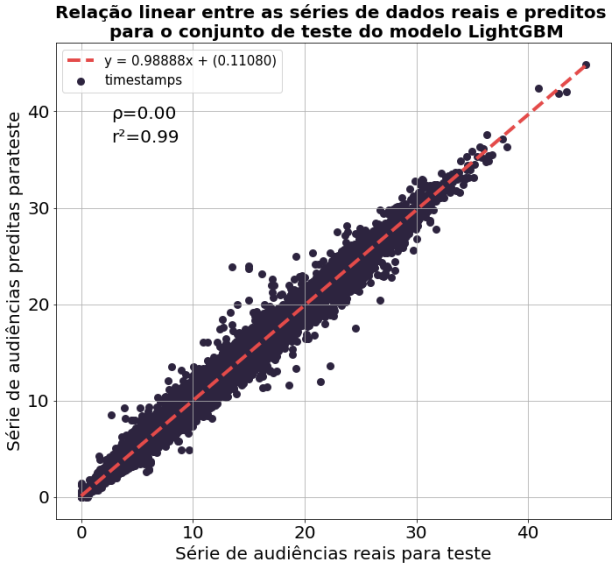


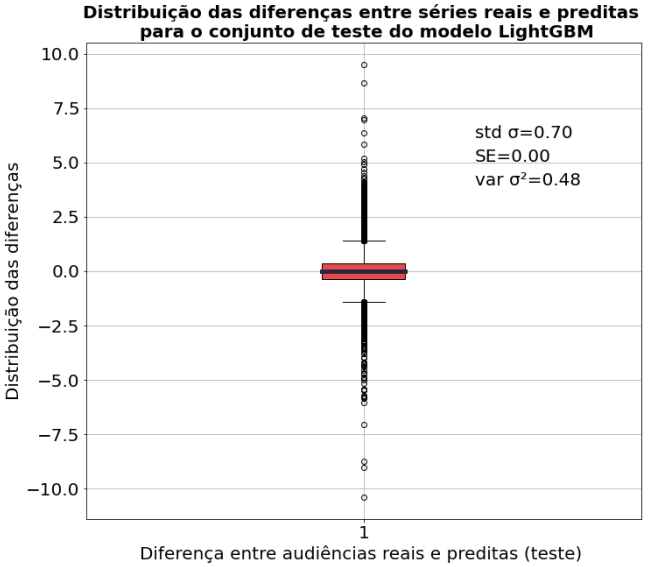


Resultados Teste

Em contrapartida, os resultados de teste mostram que o coeficiente de significância diminuiu, e passou a valer 0,99, enquanto a variância e o desvio padrão aumentaram para 0,48 e 0,70, respectivamente, e erro médio quadrático de 0,48. Contudo, o valor do erro médio padrão não consta no gráfico, mas possui o resultado de 0,48.

Percebe-se uma leve mudança na dispersão, já que os dados acabam se espalhando um pouco mais, já esperado na diferença entre resultados de treino e teste, mas é importante ressaltar que essa mudança é muito sutil, ainda sendo favorável.





Esses resultados apenas confirmam que o LGBM é um dos melhores modelos que temos, afinal, o coeficiente de determinação chega muito próximo de 1 no teste. Além disso, a variância e o desvio padrão apresentaram valores baixos, o que confirma a eficácia do modelo.

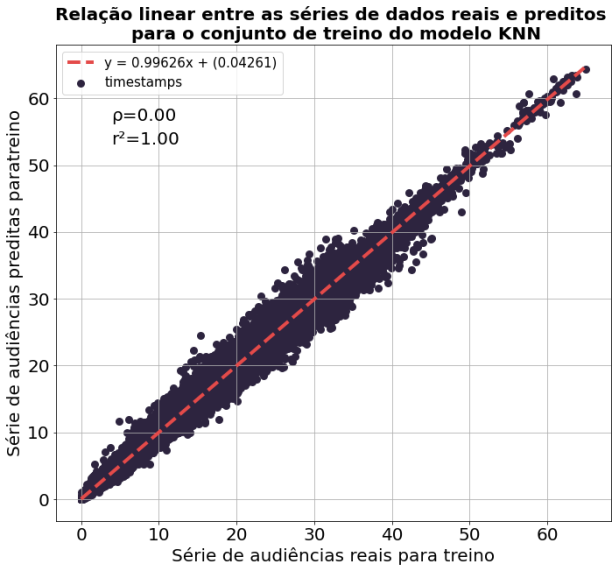
4.5.2. Resultados KNN

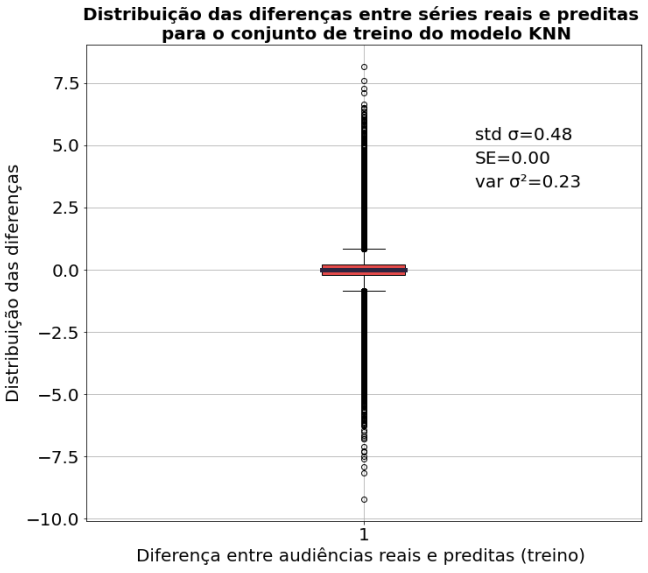
Resultados Treino

Analisando as métricas do modelo KNN, se obteve o resultado do r^2(coeficiente de determinação) dos dados de treino igual a 1, além de apresentarem uma variância e desvio padrão de 0,23 e 0,48, respectivamente, e erro médio quadrático de 0,23. Contudo, o valor do erro médio padrão não consta no gráfico, mas possui o resultado de 0,31.

Dessa forma, é possível afirmar que o KNN apresentou um ótimo resultado, pois no primeiro gráfico os dados estão próximos à linha da regressão linear. Ademais, no segundo gráfico o desvio padrão e a variância apresentaram valores próximos a zero, tendo metade de suas diferenças entre audiências reais e preditas muito próxima de zero e outliers que se encontram concentrados 6 pontos de diferença para cima e para baixo.

O modelo KNN obteve o melhor dos resultados entre os modelos que escolhemos para seguir para a fase de teste. O KNN é um dos mais dependentes da qualidade de hiper parametrização, e isso se dá por ter o K (representante do número de vizinhos próximos para a comparação) como um parâmetro. É possível visualizar nos gráficos acima que o resultado do treino foi muito pouco disperso, com seu R^2 atingindo um resultado de 1, que seria o equivalente a 100%, com quase todos os valores em cima da reta padrão. O motivo do KNN ter sido nosso melhor modelo se dá ao fato do KNN ser forte mesmo com datasets com muito ruído, pois pegam os valores que orbitam o valor imputado, raramente se espalhando à uma anomalia.

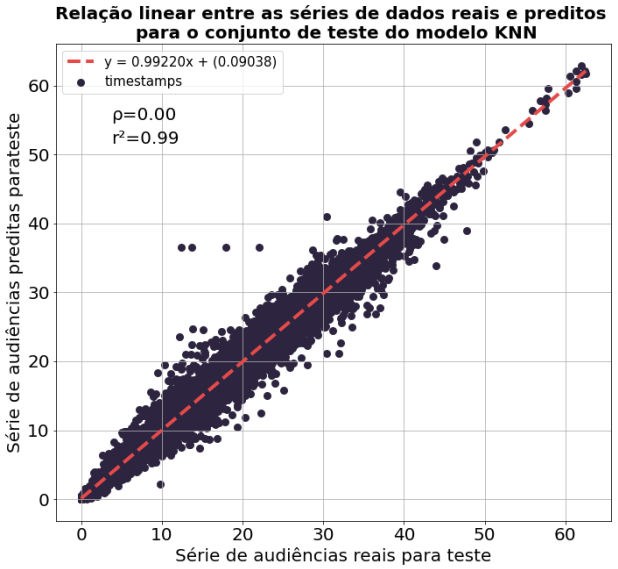


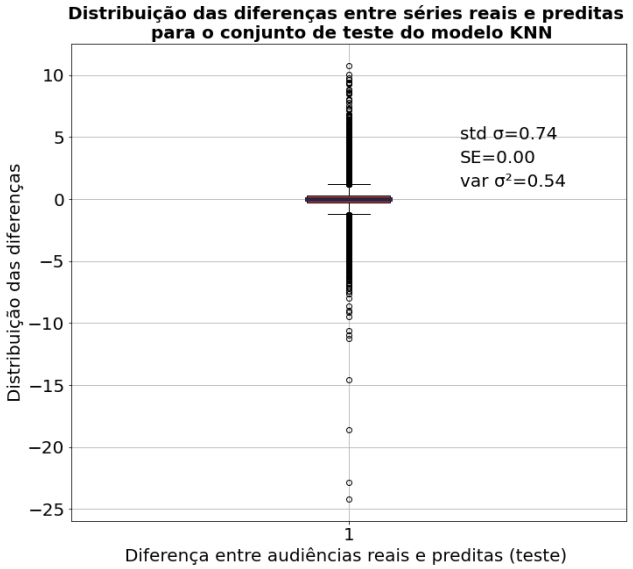


Resultados Teste

Analisando os gráficos de teste, os resultados mostram que o coeficiente de determinação cai em 0,01, para 0,99 , enquanto a variância e o desvio padrão aumentaram um pouco, para 0,54 e 0,74, respectivamente, e erro médio quadrático de 0,54. Contudo, o valor do erro médio padrão não consta no gráfico, mas possui o resultado de 0,46.

Os dados continuam alinhados com a reta, a mudança quase não é perceptível. O mesmo ocorre com o plot box, metade de seus dados continuam muito próximos de zero, mas o valor dos seus outliers é maior, tendo diferenças de 10 pontos.



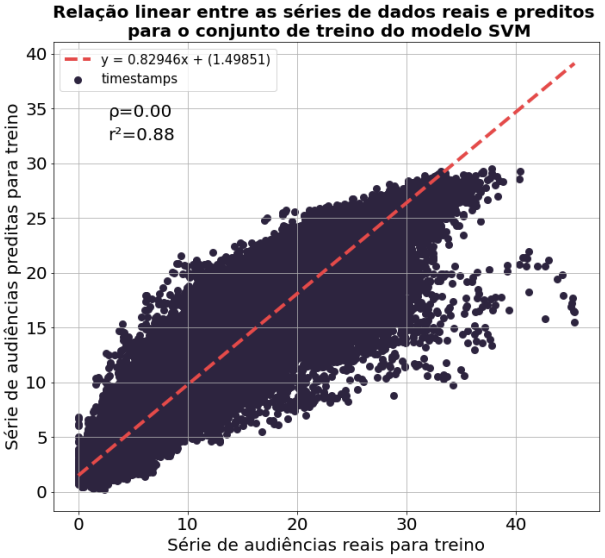


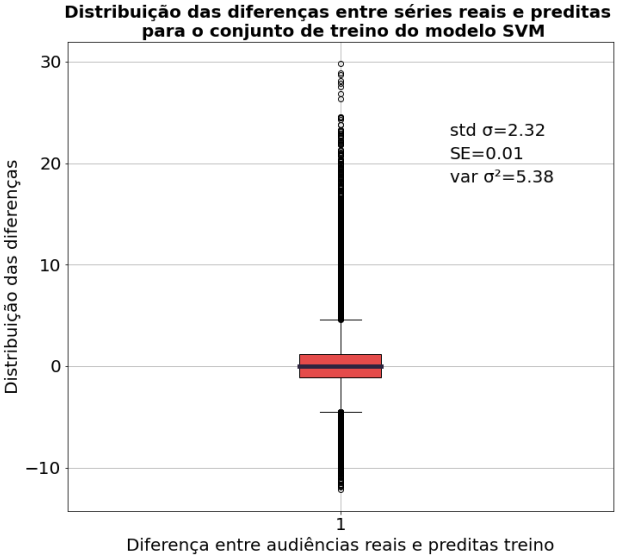
4.5.3. Resultados Support Vector Machine

Os resultados dos gráficos de treino e teste do Support Vector mostram que tanto seu treino, quanto seu teste, obtiveram resultados muito semelhantes. Isso se deve ao fato do Support Vector ser um tipo de regressão linear, as quais normalmente conseguem resultados similares em seus treinos e testes. Além disso, tivemos um desvio padrão de 4,51 e 4,52, e erro médio quadrático de 5,38 e 10,08, no treino e no teste, respectivamente.

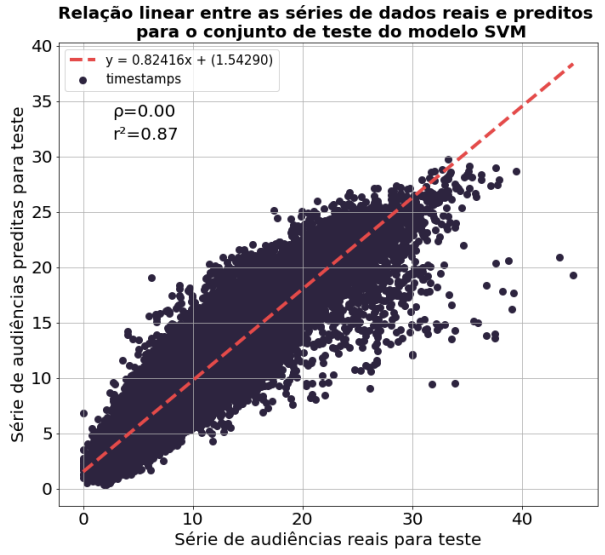
Logo, pode-se concluir que o Support Vector Machine não apresentou bons resultados, pois no primeiro gráfico fica evidente que os dados estão muito dispersos e distantes da linha da regressão linear e seu pot box apresenta uma distância maior de zero e uma quantidade elevada de outliers de valor acima de 15. Além disso, a variância e o desvio padrão apresentaram números muito altos para a utilização do modelo.

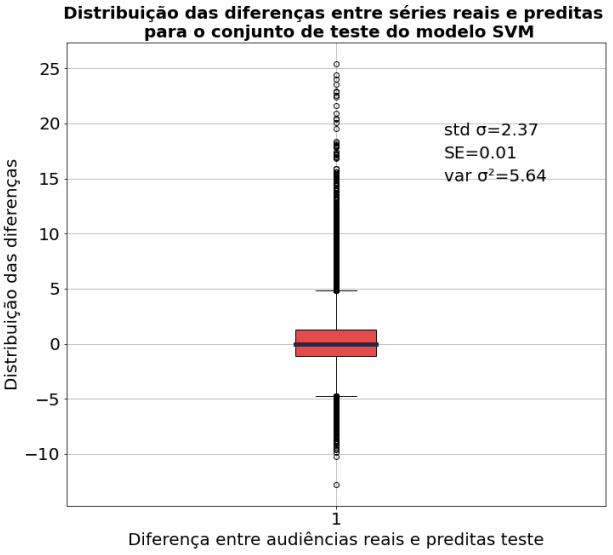
Resultados Treino





Resultados Teste



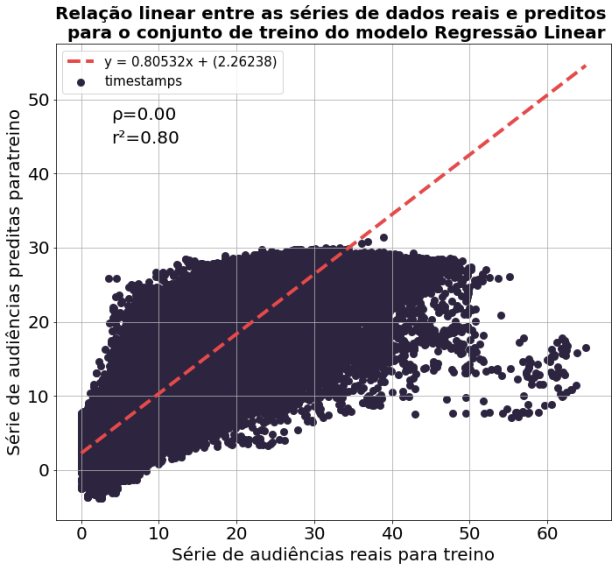


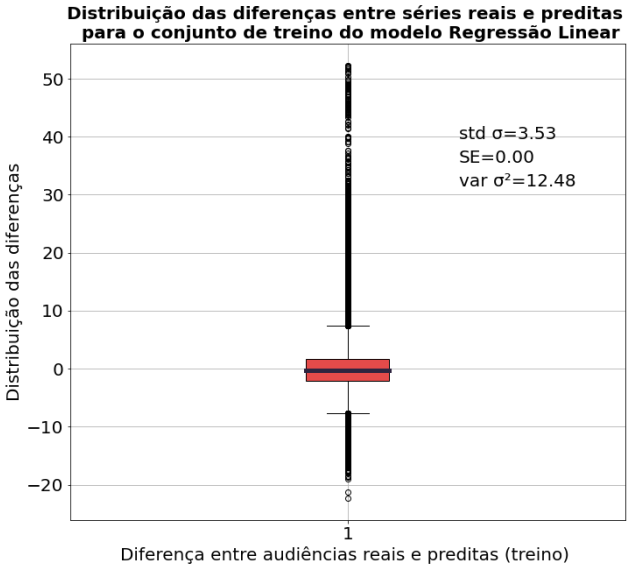
4.5.4. Resultados Regressão Linear

Resultados Treino

Os resultados de treinamento do modelo de regressão linear, demonstram um r^2(coeficiente de determinação) equivalente a 0,80, também apresentam uma variância, desvio padrão de 3,53 e erro médio quadrático de 12,48. Já o erro médio absoluto, que não está presente no gráfico, tem seu resultado em 2,53.

Dessa maneira, assim como o Support Vector Machine, a Regressão Linear também não apresentou bons resultados, pois os dados estão dispersos da linha da regressão linear, e além disso, o desvio padrão e variância apresentaram valores maiores do que o desejável. Seu box plot possue a metade dos valores mais dispersa e outliers de até 50 de diferença, sendo nenhum pouco favorável.

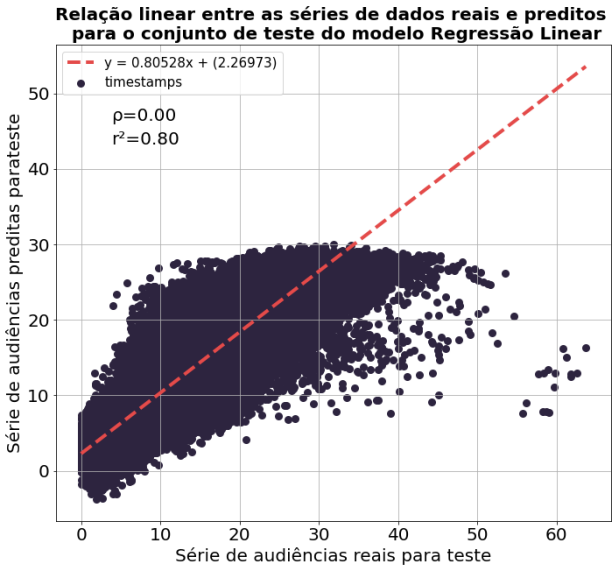


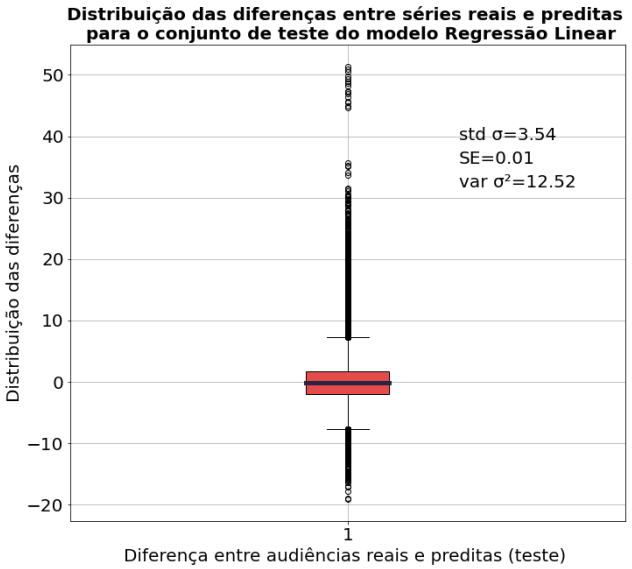


Resultados Teste

Já na análise dos gráficos de teste os resultados mostram que o coeficiente de determinação se manteve ,enquanto a variância e o desvio padrão aumentaram, passando de 3,53 para 3,54. Enquanto o erro médio quadrático tem valor de 12,52 e erro médio absoluto de 2,53.

Mantemos uma grande dispersão na regressão linear e uma mudança dos outliers, sendo mais concentrados em valores de até 30, mas ainda não sendo satisfatório.





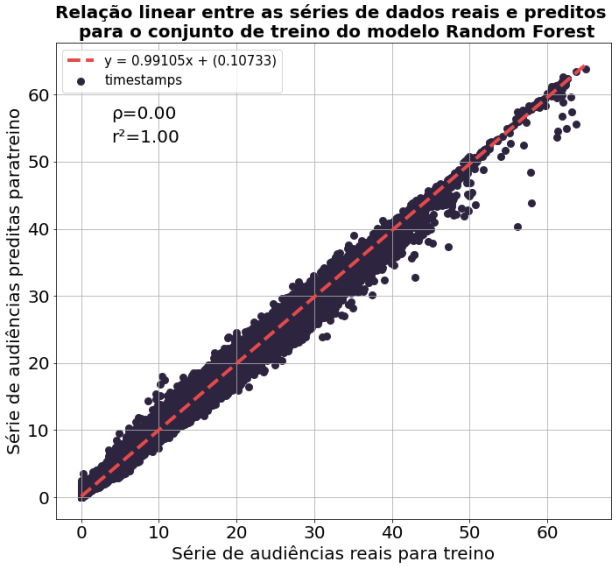
De modo geral, tanto o treino quanto o teste tiveram resultados semelhantes, sem muito sucesso.

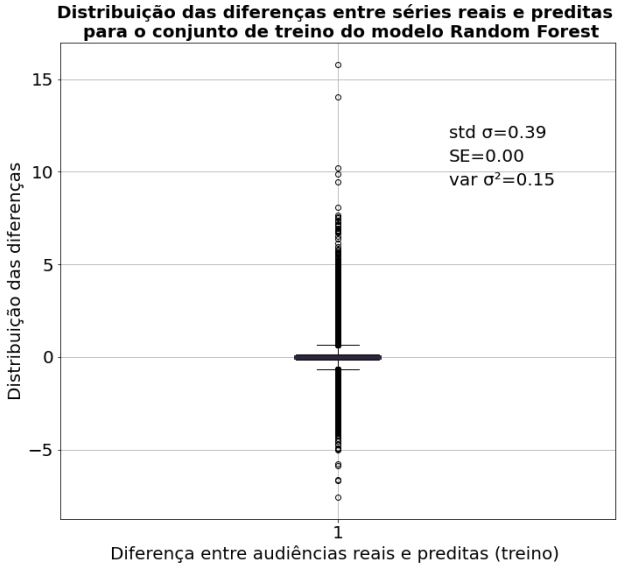
4.5.4. Resultados Random Forest

Resultados Treino

Os gráficos acima foram gerados com a intenção de analisar o modelo preditivo de Random Forest no treino e no teste. Os resultados do treino foram de um nível de significância (ρ) igual a 0,00, o r^2 (coeficiente de determinação) igual a 1, desvio padrão(σ) de 0.39 e variância(σ^2) de 0,15. A respeito das suas métricas de erro temos: erro médio quadrático em 0,15 e erro médio absoluto em 0,25.

Dessa maneira, é possível afirmar que o Random Forest apresentou ótimos resultados, pois já que o coeficiente de determinação é igual a 1, os dados se encontram muito próximos na linha de regressão linear. Sua caixa no box plot se torna uma linha de tão próximos do valor zero, um ótimo resultado, com outliers que se concentram num intervalo menor que 10 pontos.

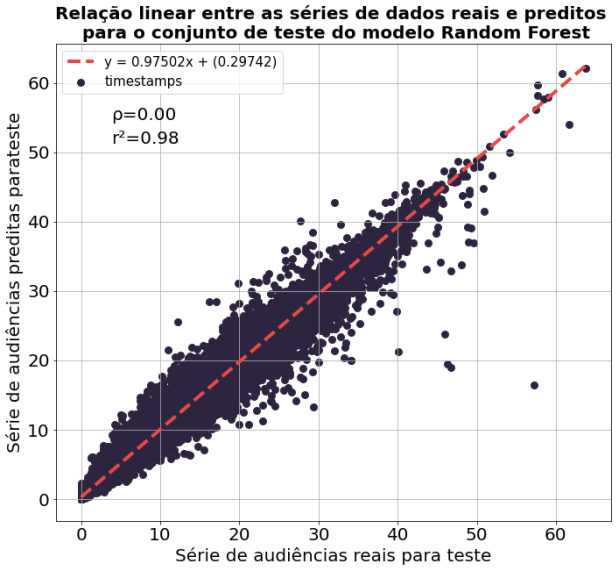


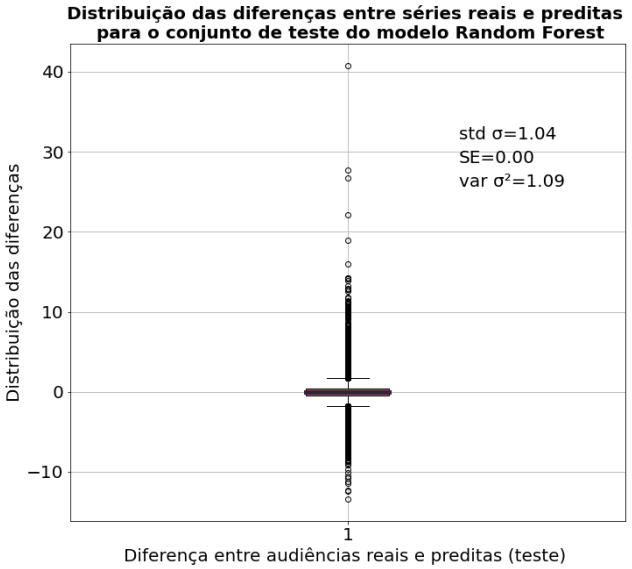


Resultados Teste

Todavia, para os resultados de teste, o de significância (ρ) se mantém igual em 0,00, o r² cai para 0,98,sofrendo assim, um decréscimo de 0,01 em relação aos dados de treino, desvio padrão(σ) aumenta para 1.04 e a variância(σ²) tem um aumento drástico para 1.09. Em suas métricas de erro temos: erro médio quadrático em 1,09 e erro médio absoluto em 0,67.

Há uma leve dispersão dos dados na regressão e um aumento do valor dos outliers, sendo mais recorrentes até 15 pontos.





4.6. Comparação de Modelos

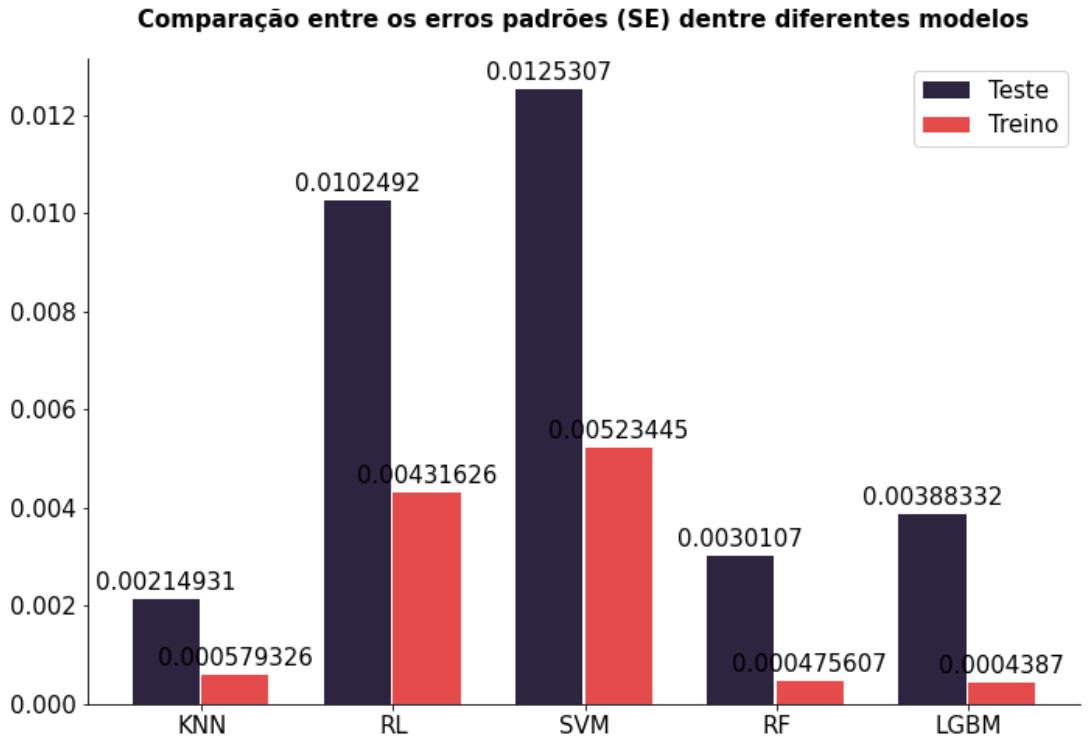
4.6.1. Comparação de resultado de métricas entre modelos

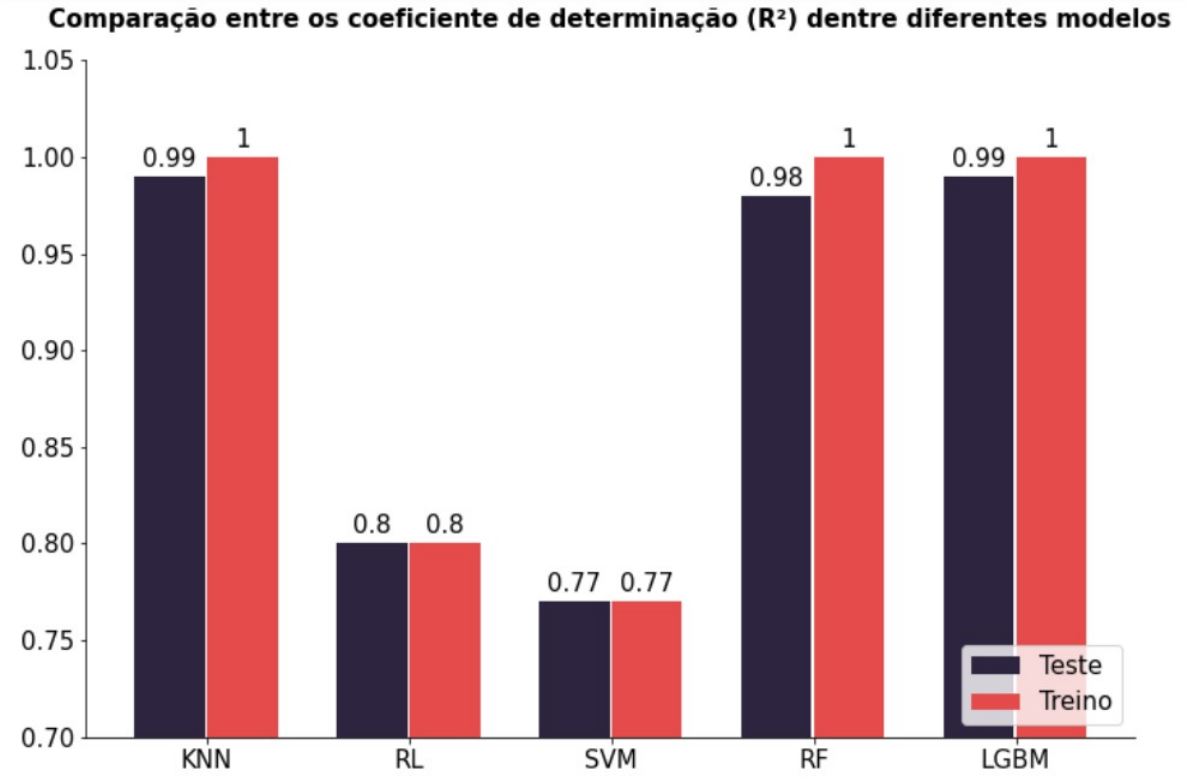
A comparação de modelos foi feita utilizando uma função para gerar gráficos de barras. Este tipo de visualização é preferível quando se deseja comparar dados de uma mesma unidade, além de servir para visualizar como seus valores estão em relação uns aos outros. Os gráficos são alimentados utilizando os outputs obtidos a partir da comparação da dispersão linear feita entre os modelos(contida no item 4.5). As métricas, significância(ρ), coeficiente de determinação(r^2), desvio padrão(σ) e variância(σ^2), foram escolhidas porque servem bem para avaliar o modelo e conseguir diferenciar seus resultados.

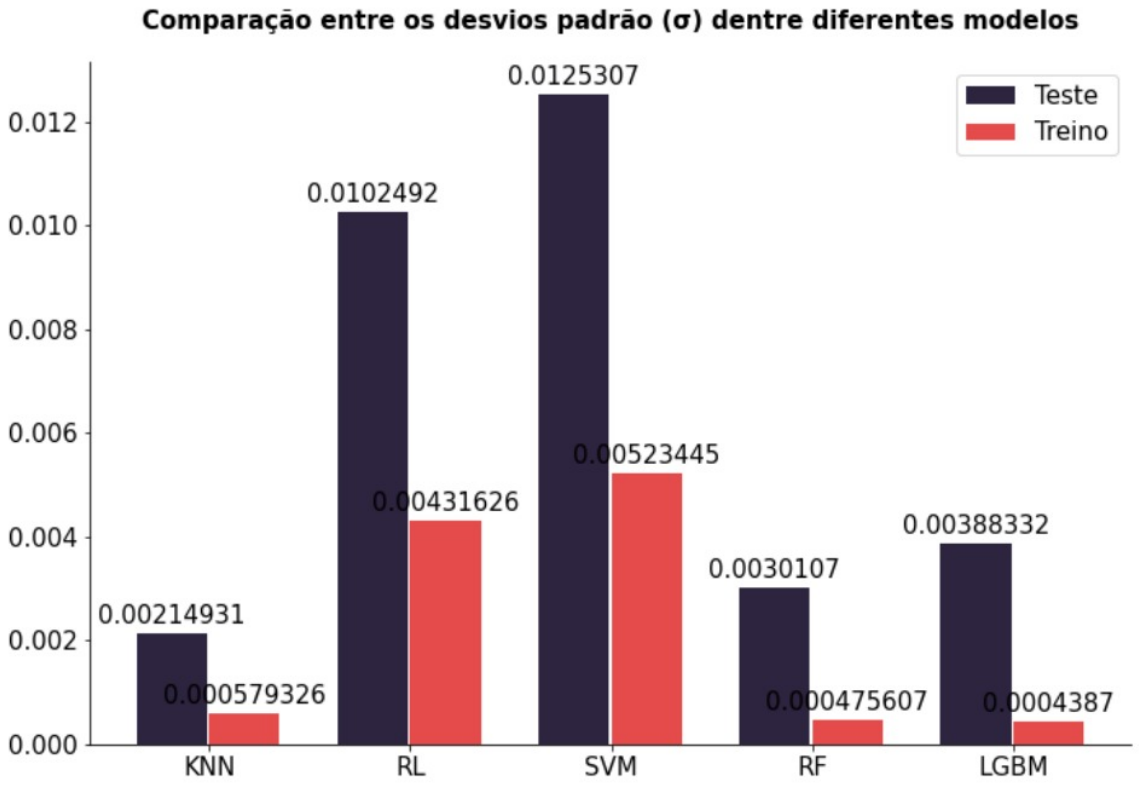
Os gráficos colocados a seguir, contêm descrições de qual métrica foi usada neles e qual modelo obteve o melhor resultado. Essa comparação de modelos foi realizada visando determinar o melhor modelo com base nessas métricas e o porquê dele ser sido eleito como o melhor.

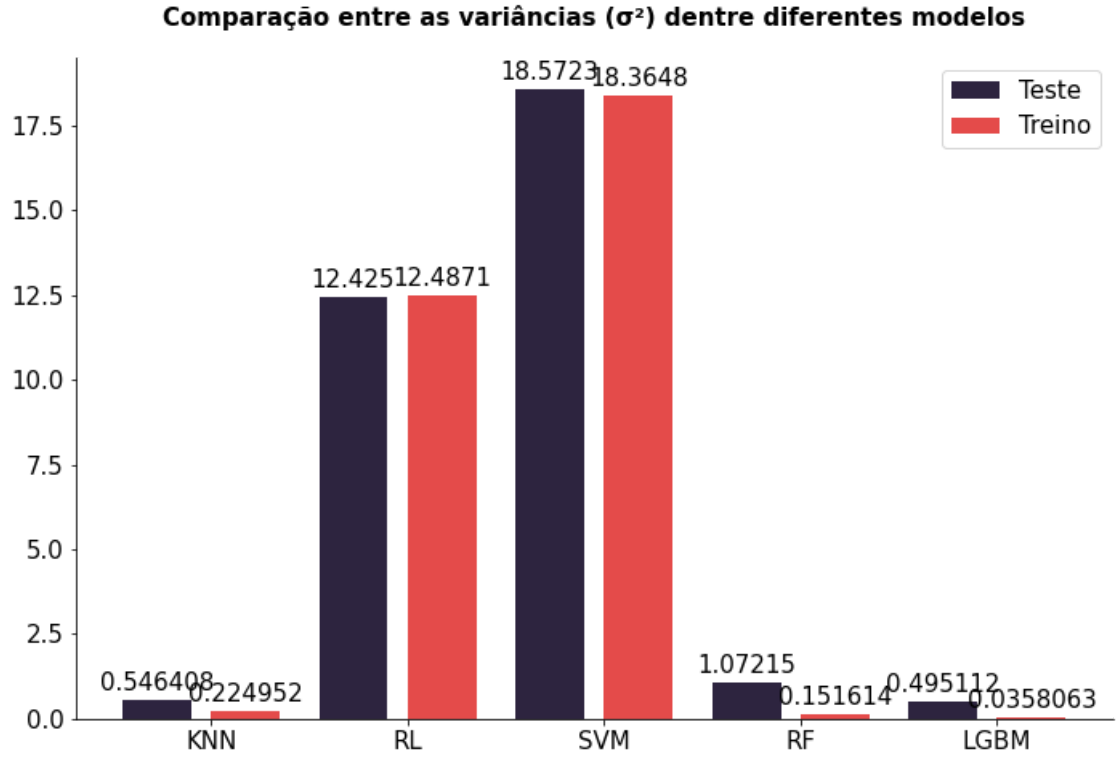
Com base nesses gráficos, é possível analisar que os modelos KNN e LGBM apresentaram os melhores resultados, pois eles possuem o r^2 próximo de 1. Além disso, a variância e o desvio padrão desses modelos apresentam valores baixos, o que é excelente para a performance dos modelos.

Por outro lado, a regressão linear e o SVM obtiveram os piores resultados, pois possuem o r^2 mais próximo de zero. Ademais, esses dois modelos foram os que apresentaram os maiores valores no desvio padrão e variância, logo eles não serão utilizados na versão final do projeto.


Comparação entre os erros padrões (SE) dentre diferentes modelos
Teste
Treino
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0.00214931
0.000579326
0.0102492
0.00431626
0.0125307
0.00523445
0.0030107
0.000475607
0.00388332
0.0004387
KNN
RL
SVM
RF
LGBM

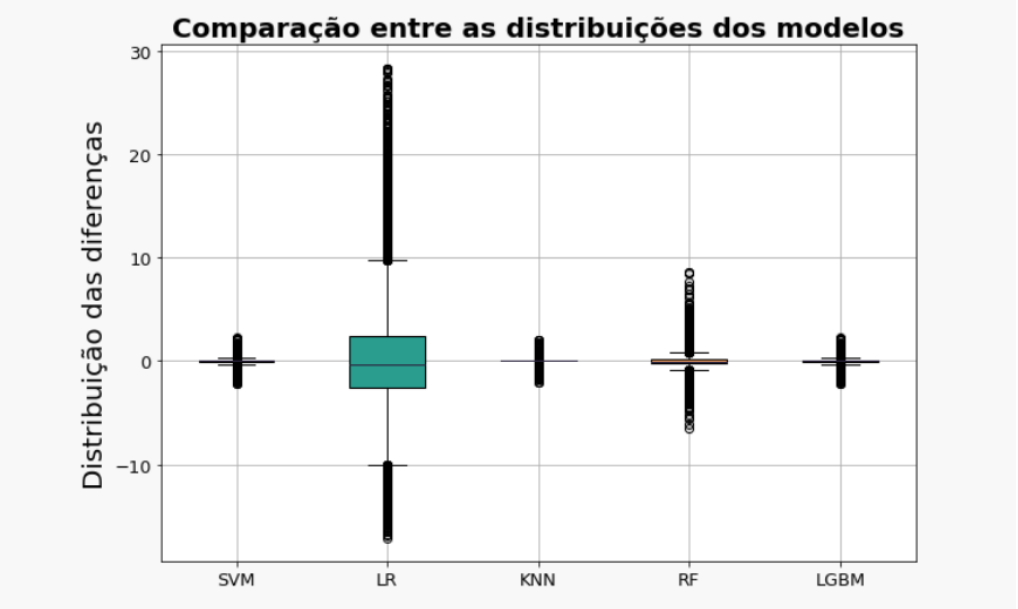

Comparação entre os coeficiente de determinação (R²) dentre diferentes modelos
1.05
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.99
1
0.8
0.8
0.77
0.77
0.98
1
0.99
1
Teste
Treino
KNN
RL
SVM
RF
LGBM


Comparação entre os desvios padrão (σ) dentre diferentes modelos
Teste
Treino
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0.00214931
0.000579326
0.0102492
0.00431626
0.0125307
0.00523445
0.0030107
0.000475607
0.00388332
0.0004387
KNN
RL
SVM
RF
LGBM


Comparação entre as variâncias (σ²) dentre diferentes modelos
Teste
Treino
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.546408
0.224952
12.4251
12.4871
18.5723
18.3648
1.07215
0.151614
0.495112
0.0358063
KNN
RL
SVM
RF
LGBM

A seguir, há a comparação entre os gráficos de box plot de todos os modelos. Para melhor entendimento, é necessário saber que a caixa encontrada ao meio representa a metade dos dados analisados e onde eles se encontram, então, quanto menor a caixinha de um modelo, mais próximo de zero estão a metade das diferenças entre os dados reais e previstos, logo, um modelo mais assertivo. Os pontinhos representam a localização dos outliers.

Logo, concluí-se que o KNN é o que possui a menor diferença entre dados reais e previstos em sua metade, tendo sua caixinha tão fina que se tornou uma linha e uma dispersão de outliers que não atinge nem 5 pontos, logo, uma diferença muito pequena.



4.6.2. Comparação de modelos utilizando Auto Machine Learning

A comparação de modelos utilizando Auto Machine Learning foi feita com a biblioteca de Python, Pycaret. Essa biblioteca é muito útil quando não se tem ideia de quais são os modelos que mais se ajustam ao DataFrame usado, o que não foi o caso, visto a do projeto no momento. Ele cria modelos diferentes e compara suas principais métricas para chegar ao melhor modelo sem precisar percorrer muitas etapas anteriores.

Portanto, é de suma importância explicar que, mesmo que os resultados não tenham sido satisfatórios para o modelo escolhido, KNN, o grupo decidiu por manter esse como o modelo definitivo. O resultado não satisfatório se deu por conta da manipulação dos dados que, por mais que seja uma boa biblioteca para manipulação de dados, não possui todas as manipulações de dados necessárias para gerar uma precisão muito alta.

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
rf	Random Forest Regressor	9.477000e-01	1.762200e+00	1.327500e+00	9.585000e-01	0.1393	1.283000e-01	188.960
et	Extra Trees Regressor	1.059100e+00	2.262900e+00	1.504300e+00	9.467000e-01	0.1573	1.414000e-01	161.575
knn	K Neighbors Regressor	1.178500e+00	2.891800e+00	1.700500e+00	9.319000e-01	0.1633	1.524000e-01	199.095
dt	Decision Tree Regressor	1.188000e+00	3.134900e+00	1.770400e+00	9.262000e-01	0.1819	1.540000e-01	6.470
lightgbm	Light Gradient Boosting Machine	1.495500e+00	3.978200e+00	1.994600e+00	9.063000e-01	0.2170	2.234000e-01	2.055
gbr	Gradient Boosting Regressor	1.830700e+00	6.155200e+00	2.481000e+00	8.551000e-01	0.2556	2.757000e-01	37.900
ridge	Ridge Regression	1.968500e+00	7.086500e+00	2.662000e+00	8.331000e-01	0.2791	2.825000e-01	1.390
br	Bayesian Ridge	1.968300e+00	7.085600e+00	2.661900e+00	8.331000e-01	0.2790	2.824000e-01	15.260
lr	Linear Regression	1.968000e+00	7.094200e+00	2.663500e+00	8.329000e-01	0.2789	2.821000e-01	9.835
omp	Orthogonal Matching Pursuit	2.163200e+00	8.401000e+00	2.898400e+00	8.022000e-01	0.3078	3.229000e-01	1.815
huber	Huber Regressor	2.112100e+00	8.432100e+00	2.903800e+00	8.014000e-01	0.2975	3.056000e-01	88.075
par	Passive Aggressive Regressor	2.129600e+00	8.668600e+00	2.944100e+00	7.959000e-01	0.2871	3.000000e-01	15.095
ada	AdaBoost Regressor	3.454400e+00	1.708290e+01	4.133100e+00	5.977000e-01	0.5050	7.548000e-01	77.190

lr	Linear Regression	1.968000e+00	7.094200e+00	2.663500e+00	8.329000e-01	0.2789	2.821000e-01	9.8
omp	Orthogonal Matching Pursuit	2.163200e+00	8.401000e+00	2.898400e+00	8.022000e-01	0.3078	3.229000e-01	1.815
huber	Huber Regressor	2.112100e+00	8.432100e+00	2.903800e+00	8.014000e-01	0.2975	3.056000e-01	88.075
par	Passive Aggressive Regressor	2.129600e+00	8.668600e+00	2.944100e+00	7.959000e-01	0.2871	3.000000e-01	15.095
ada	AdaBoost Regressor	3.454400e+00	1.708290e+01	4.133100e+00	5.977000e-01	0.5050	7.548000e-01	77.190
en	Elastic Net	3.255100e+00	1.860500e+01	4.313300e+00	5.619000e-01	0.4410	4.948000e-01	1.080
lasso	Lasso Regression	3.258700e+00	1.863920e+01	4.317300e+00	5.611000e-01	0.4405	4.961000e-01	1.320
llar	Lasso Least Angle Regression	5.226300e+00	4.246480e+01	6.516500e+00	-0.000000e+00	0.6533	1.006000e+00	1.965
dummy	Dummy Regressor	5.226300e+00	4.246480e+01	6.516500e+00	-0.000000e+00	0.6533	1.006000e+00	0.285
lar	Least Angle Regression	4.541921e+09	4.703371e+23	5.207307e+11	-1.109405e+22	15.9693	4.393599e+08	2.310

```
INFO:logs:create_model_container: 18
INFO:logs:master_model_container: 18
INFO:logs:display_container: 2
INFO:logs:RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
```

5. Conclusões e Recomendações

Dessa forma, pode-se concluir que o grupo TvCoders testou cinco modelos de regressão: LGBM, Regressão Linear, KNN, Random Forest e Support Vector Machine. Entretanto, nem todos apresentaram bons resultados. Logo, o modelo escolhido para o PredTV foi o KNN, pois de acordo com os gráficos mostrados na seção 4.6, ele obteve os melhores resultados.

Todavia, como qualquer modelo preditivo, o PredTv corre o risco de ter algum dado enviesado e isso prejudicar os resultados de audiência que a emissora deseja. Portanto, é importante que a Rede Gazeta atualize o dataset a cada ano, para que o modelo treine os novos dados e possíveis mudanças.

6. Referências

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

(1) https://agenciabrasil.ebc.com.br/geral/noticia/2022-01/tv-brasil-avanca-e-ja-e-5a-emissora-mais-assistida-do-pais

(2) https://www.poder360.com.br/midia/globo-registra-receita-de-r-144-bi-e-prejuizo-de-r-173-mi/#:~:text=A%20Globo%20Comunica%C3%A7%C3%A3o%20e%20Participa%C3%A7%C3%B5es,a%20n%C3%ADveis%20anteriores%20%C3%A0%20pandemia.

(3) https://noticiasdatv.uol.com.br/noticia/mercado/em-crise-no-ibope-sbt-lucra-r-141-milhoes-gracas-futebol-e-show-do-milhao-81153

(4) https://noticiasdatv.uol.com.br/noticia/mercado/record-investe-r-622-milhoes-mas-tem-lucro-menor-que-o-sbt-em-2021-81476

(5) https://oglobo.globo.com/economia/negocios/noticia/2022/05/tv-aberta-e-canais-por-assinatura-concentram-79percent-do-consumo-de-video-do-brasileiro.ghtml

(6) https://www.uol.com.br/splash/noticias/ooops/2022/02/04/veja-o-ranking-de-ibope-da-tv-aberta-redetv-ja-ronda-o-traco.htm

Anexos

A seguir, o manual do usuário, para que o usuário possa entender como utilizar a interface gráfica que foi desenvolvida:

https://www.canva.com/design/DAFOGwrWVPY/vkM809XsfiLjjMOpbCcRaA/view?utm_content=DAFOGwrWVPY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton