Análise de Dados – Exploratória e ML

Sumário

Análise de Dados para *Let's Snack*
Socialment

Arthur Micarelli Domingos
Enzo Yudi de Oliveira Hino

Introdução

→ o problema que tenta resolver

Nosso aplicativo é projetado para transformar a maneira como as pessoas interagem com a alimentação, abordando o problema central de como as preferências pessoais, restrições alimentares e a relação emocional com a comida podem complicar a escolha de receitas e o planejamento de refeições.

→ o público-alvo

Nosso público-alvo é composto por indivíduos conscientes da importância da alimentação para o bem-estar e interessados em estilos de vida saudáveis. Eles buscam informações sobre nutrição, hábitos saudáveis e estão motivados a fazer escolhas alimentares mais saudáveis. Além disso, esse público pode incluir adeptos de dietas específicas, como veganismo e vegetarianismo, que também estão preocupados com o meio ambiente e buscam opções alimentares alternativas.

Objetivo do aplicativo

Para resolver o problema descrito, o aplicativo reúne em um único lugar várias funcionalidades que facilitam a busca por receitas adequadas, eliminando a necessidade de consultar múltiplas fontes. Ele também oferece opções personalizadas de receitas que respeitam as necessidades alimentares específicas de cada usuário. Além disso, o aplicativo sugere pratos com base nos ingredientes disponíveis, ajudando a evitar desperdícios. Por fim, ele organiza o processo de planejamento de refeições ao permitir que os usuários salvem suas receitas favoritas e criem listas de compras de forma prática. Assim, o aplicativo torna o planejamento alimentar mais simples e acessível, contribuindo para uma alimentação mais equilibrada e adequada às necessidades individuais.

Objetivo da Análise Exploratória de Dados

A análise exploratória de dados tem como objetivo, neste projeto, auxiliar no entendimento dos dados adquiridos e validar suas informações. Além disso, é de grande importância para identificar pontos de melhoria e alinhar da melhor forma a base com os objetivos do projeto. Isso é perceptível na nossa produção da IA, pois, graças à análise exploratória, foi possível identificar que parte da base foi gerada, e assim, essa parte dos dados não estava de acordo com a lógica do projeto, como, por exemplo, a idade em formato float.

Objetivo da Análise de Dados (para Aprendizado de Máquina)

O objetivo da análise de dados é a aplicação dos modelos de machine learning aprendidos em sala. Por isso, foram utilizados, para as análises, os algoritmos: Naive Bayes, KNN e Decision Tree.

O modelo escolhido, junto com outro modelo não supervisionado, trará como resultado uma predição para um potencial cliente do APP, de acordo com a nossa justificativa de uso.

Levantamento dos dados na análise exploratória

Busca dos dados

A busca por dados foi realizada na web, com o objetivo de encontrar datasets que se relacionassem com o aplicativo desenvolvido e com a experiência das pessoas em relação à alimentação. As buscas mais relevantes foram realizadas em repositórios do GitHub, utilizando termos como “Datasets de Hábitos Alimentares”.

Os critérios de seleção foram definidos de forma a garantir a confiabilidade das fontes e a relevância dos dados para a predição de possíveis usuários. Dessa forma, foi fundamental que o dataset possuísse uma fonte confiável e estivesse relacionado ao comportamento alimentar e suas implicações.

Os datasets que melhor atenderam aos critérios estabelecidos foram:

1. **Estimation of Obesity Levels Based On Eating Habits and Physical Condition**
    - Repositório: GitHub
    - Fonte: UCI Machine Learning Repository
2. **Eating & Health Module Dataset**
    - Repositório: GitHub
    - Fonte: Kaggle

Justificativa de uso

Nosso público-alvo é composto por indivíduos que valorizam a importância da alimentação para o bem-estar e estão interessados em manter um estilo de vida saudável. Por isso, buscamos bases de dados que nos permitissem analisar informações sobre pessoas com diferentes perfis, tanto aquelas que se alinham ao nosso público-alvo quanto aquelas que não. Isso é essencial para conseguirmos identificar e diferenciar características específicas entre esses grupos.

As bases de dados selecionadas foram escolhidas por oferecerem a melhor qualidade de informações sobre as relações das pessoas com o nosso aplicativo.

A base de dados “**Eating & Health Module Dataset**” foi escolhida por conter um estudo abrangente sobre hábitos alimentares, o que nos permite categorizar as pessoas com base em seus hábitos alimentares, sejam eles saudáveis ou não. Isso oferece uma base sólida para o treinamento da nossa inteligência artificial. No entanto, essa base não possui uma variável de resposta (coluna Y), o que limita o uso de métodos tradicionais de Machine Learning.

Já a base “**Estimation of Obesity Levels Based On Eating Habits and Physical Condition**” possui uma variável de resposta (coluna Y) relacionada ao peso das pessoas, sendo especialmente útil para a classificação de níveis de peso. Com essa base, podemos concluir que usuários com níveis de gordura corporal fora do padrão, provavelmente não compartilham dos hábitos alimentares saudáveis que caracterizam o nosso público-alvo. No entanto, é importante ressaltar que a intenção não é excluir ou discriminar esses indivíduos, mas sim entender melhor os diferentes perfis de usuários para oferecer soluções mais adequadas ao nosso público-alvo.

Descrição da base de dados de trabalho

77% dos dados foram gerados sinteticamente usando a ferramenta Weka e o filtro SMOTE, 23% dos dados foram coletados diretamente dos usuários por meio de uma plataforma web.

Logo, as colunas: [FCVC, NCP, CH2O, FAF, TUE] começam como int, por serem dados qualitativos, mas ao gerar os dados foram inseridos floats. Além disso, a coluna "Age", mesmo não sendo qualitativa, veio como float, não sendo adequado.

Segue a descrição da base, de como ela veio, não como a documentação original descreve.

1. **Estimation of Obesity Levels Based On Eating Habits and Physical Condition**

Nome das Variaveis	Tipo	Descrição
Gender	String	Gênero
Age	Float	Idade
Height	Float	Altura
Weight	Float	Peso
family_history_with_overweight	Boolean	Algum membro da família sofreu ou sofre de sobrepeso?
FAVC	Boolean	Você consome alimentos ricos em calorias com frequência?
FCVC	Float	Você geralmente consome vegetais nas suas refeições?
NCP	Float	Quantas refeições principais você faz diariamente?
CAEC	String	Você consome qualquer alimento entre as refeições?
SMOKE	Boolean	Você fuma?
CH2O	Float	Quantos copos de água você bebe diariamente?
SCC	Boolean	Você monitora as calorias que consome diariamente?
FAF	Float	Com que frequência você realiza atividade física?
TUE	Float	Quanto tempo você passa usando dispositivos tecnológicos, como celular, videogames, TV?
CALC	String	Com que frequência você consome álcool?
MTRANS	String	Qual meio de transporte você utiliza com mais frequência?

Nome das Variaveis	Tipo	Descrição
NObeyesdad	String	Nível de obesidade

2. **Eating & Health Module Dataset**

Nome das Variáveis	Tipo	Descrição
tucaseid	int	Identificador único para cada caso ou respondente.
tulineno	int	Número da linha do respondente dentro de um caso específico.
eeincome1	float	Faixa de renda do respondente.
erbmi	float	Índice de Massa Corporal (IMC) do respondente.
erhhch	int	Número de membros no domicílio.
erincome	float	Renda total do domicílio.
erspemch	float	Despesa mensal com alimentos.
ertpreat	float	Tempo gasto comendo antes da entrevista (em minutos).
ertseat	float	Tempo gasto comendo durante a entrevista (em minutos).
ethgt	float	Altura do respondente.
etwgt	float	Peso do respondente.
eudietsoda	int	Frequência de consumo de refrigerante diet.
eudrink	int	Frequência de consumo de bebidas alcoólicas.
eueat	int	Frequência de refeições.
euexercise	int	Frequência de exercícios físicos.
euexfreq	int	Frequência de exercícios por semana.
eufastfd	int	Frequência de consumo de fast food.
eufastfdfrq	int	Frequência semanal de consumo de fast food.
euffyday	int	Frequência de consumo de frutas.
eufdsit	int	Situação alimentar durante a entrevista.
eufinlwgt	float	Peso final atribuído ao respondente.
eusnap	boolean	Participação no programa SNAP (assistência nutricional).
eugenhth	int	Autoavaliação da saúde geral.
eugroshp	int	Frequência de compras de supermercado.
euhgt	float	Altura medida do respondente.
euinclvl	int	Nível de renda do respondente.
euincome2	float	Renda adicional.
eumeat	int	Frequência de consumo de carne.
eumilk	int	Frequência de consumo de leite.
euprpmel	int	Frequência de preparação de refeições.

Nome das Variáveis	Tipo	Descrição
eusoda	int	Frequência de consumo de refrigerantes.
eustores	int	Número de lojas visitadas para compras de alimentos.
eustreason	int	Motivo principal para a escolha de uma loja.
eutherm	int	Uso de termômetro na preparação de alimentos.
euwgt	float	Peso registrado do respondente.
euwic	boolean	Participação no programa WIC (Programa Especial de Nutrição Suplementar para Mulheres, Bebês e Crianças).
exincome1	float	Renda extra ou adicional não especificada.

Parâmetros estatísticos

Colunas Quantitativos:

Média:

- Age → 23.972524869729988
- Height → 1.7016773533870204
- Weight → 86.58605812648035

Mediana:

- Age → 22.0
- Height → 1.700499
- Weight → 83.0

Moda:

- Age → 21
- Height → 1.7
- Weight → 80.0

Desvio Padrão:

- Age → 6.3086642611136226
- Height → 0.09330481986792007
- Weight → 26.1911717452047

Colunas Qualitativas:

Moda:

- Gender → “Male”
- family_history_with_overweight → “yes”

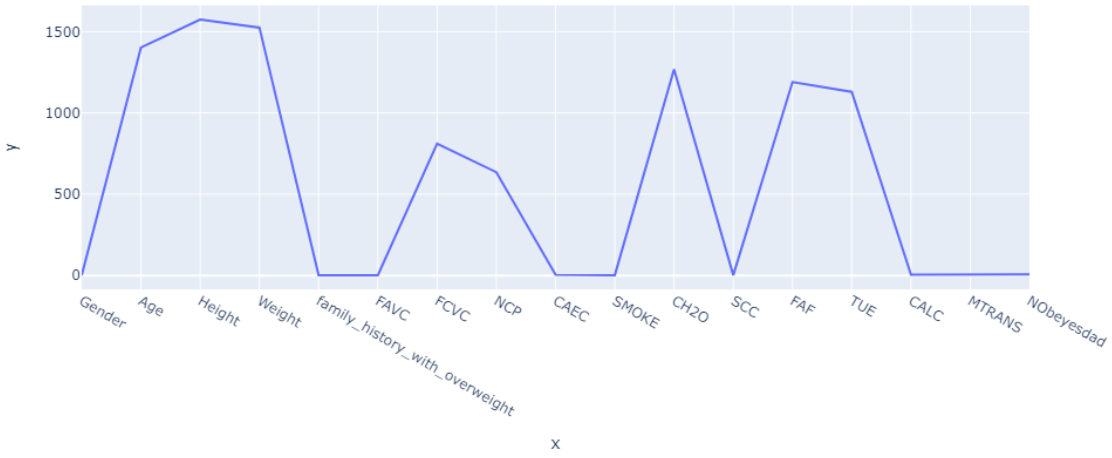
- FAVC → “yes”
- FCVC → 2
- NCP → 3
- CAEC → “Sometimes”
- SMOKE → “no”
- CH2O → 2
- SCC → “no”
- FAF → 1
- TUE → 0
- CALC → “Sometimes”
- MTRANS → “Public_Transportation”
- NObeyesdad → “Obesity_Type_I”

Todas as informações dos parâmetros foram retiradas por meio de métodos do próprio python.

Para mais informações sobre os dados quantitativos e qualitativos foram feitas diversas plotagens no arquivo “analise_exploratoria\analise_exploratoria_obesity.ipynb”, no tópico “Análise dos Dados”

Limpeza dos dados

Para iniciarmos a limpeza dos dados, foi feito um gráfico de linha, feito antes das modificações na base. O gráfico tem o intuito de mostrar quantos valores únicos as colunas qualitativas tinham antes e depois de ajustar a base.

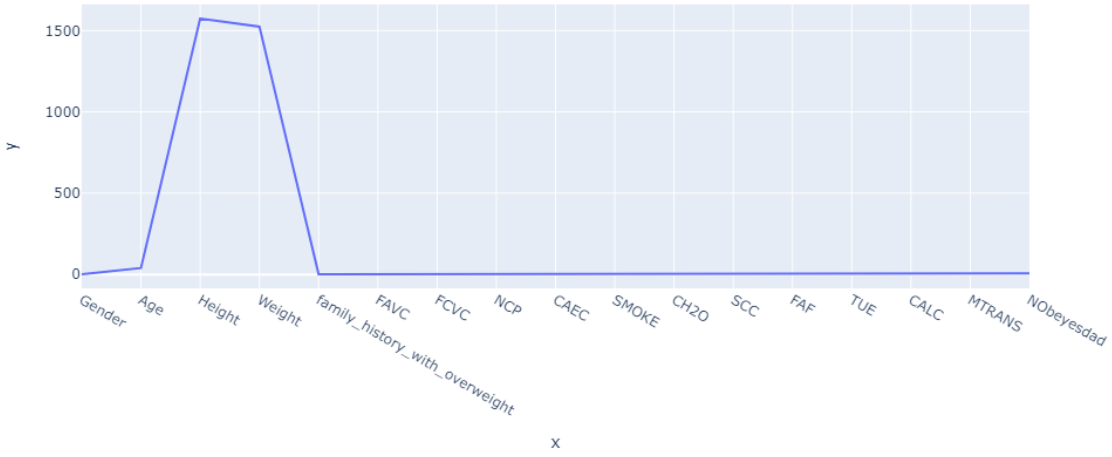


É possível perceber que as seguintes colunas: [AGE, FCVC, NCP, CH2O, FAF, TUE] não estão de acordo com a descrição original da base. Portanto, irei alterar seus valores para que se adequem ao padrão.

Os dados da coluna 'Age' estavam no formato float e continham valores decimais. Arredondamos os valores para baixo e os transformamos em inteiros.

Para arrumar os dados eu arredondei a coluna “AGE” para baixo (usando a função “floor()” da biblioteca “math”), e as demais colunas foram arredondadas para o inteiro mais próximo (“mean()”).

Após a limpeza dos dados eles ficaram assim:



Condicionamento para alimentar o modelo de ML

O processo de validação e produção dos modelos de machine learning seguiu o seguinte procedimento. Foi realizado o "Teste Simples" com todos os modelos, onde o teste foi realizado com a base padrão tratada em análise exploratória e validada com o método "cross_validate", para termos noção do desempenho médio do modelo e para prevenir overfitting. São analisados os resultados do modelo com uma série de estudos, e, por fim, foi realizado o "Teste do Modelo Baseado Nos Resultados Anteriores e na Busca pelo Melhor Desempenho Possível do Modelo". Esse teste foi feito com 5 bases diferentes: uma base é a padrão, e as outras 4 são variações da base padrão, onde os outliers da coluna "Age" foram tratados de maneiras diferentes. Os métodos utilizados foram: IQR, capping, transformação logarítmica e transformação pela raiz quadrada. A utilização de cada um desses métodos é explicada detalhadamente no arquivo “analise_exploratoria\analise_exploratoria_obesity.ipynb”, no tópico “Outliers”. Além disso, é aplicada uma pipeline de machine learning, que será utilizada para organizar os passos de transformação de dados e para aplicar o melhor modelo possível nas condições do nosso caso. Além disso, ela também é muito útil para replicar o processo em outros modelos. Cada modelo possui sua própria pipeline com os hiperparâmetros ajustados para cada caso, porém, algumas bibliotecas são usadas como padrão em todas, e essas são:

- **StandardScaler** -> Centraliza os dados em torno de zero, subtraindo a média de cada coluna e dividindo pelo desvio padrão, normalizando a dispersão dos dados.

- **SelectKBest** -> Seleciona as features mais importantes da base de dados, auxiliando na redução de dimensionalidade. Utilizamos esse método para reduzir o número de perguntas que serão feitas ao usuário, de 16 para, no máximo, 10.
- **PCA** -> Reduz a dimensionalidade após o SelectKBest, preservando o máximo de variância nos dados.
- **GridSearchCV** -> Executa a pipeline com diferentes combinações de hiperparâmetros, realiza validação cruzada e encontra a combinação que otimiza a precisão do modelo.

Para encontrar explicações mais aprofundadas sobre a pipeline, consulte os arquivos que possuem os nomes dos modelos .ipynb, na pasta "analise_de_dados".

Por fim, são analisados os resultados da pipeline (métricas, matriz de componentes do PCA e matriz de confusão) com as 5 bases.

Definição dos objetivos e das classes

Objetivos:

O principal objetivo é utilizar modelos de Machine Learning para prever se um usuário é um potencial usuário do aplicativo, ou seja, se ele se alinha ao perfil do público-alvo. O público-alvo é composto por pessoas com hábitos alimentares saudáveis, interesse em nutrição, e que seguem dietas específicas como vegetarianismo e veganismo, além de valorizarem escolhas sustentáveis.

Respostas (Variável y) e Classes:

A variável de resposta (y) será utilizada para prever o nível de peso do usuário, baseado na base de dados "Estimation of Obesity Levels Based On Eating Habits and Physical Condition". Esta variável categoriza os usuários em diferentes classes de peso, que refletem seu estado nutricional e de saúde física. As classes são:

Obesity Type I: Obesidade de tipo I.

Obesity Type II: Obesidade de tipo II.

Obesity Type III: Obesidade de tipo III (mais severa).

Overweight Level I: Sobrepeso leve.

Overweight Level II: Sobrepeso mais acentuado.

Normal Weight: Peso normal e saudável.

Insufficient Weight: Peso abaixo do recomendado.

Essas classificações são úteis para identificar o quão próximo ou distante o usuário está dos hábitos saudáveis que caracterizam o público-alvo do aplicativo. Usuários com peso normal são mais alinhados ao perfil desejado, enquanto aqueles com sobrepeso ou obesidade provavelmente estão mais distantes do nosso público-alvo.

Além do modelo supervisionado para prever o nível de peso, será utilizado um modelo não supervisionado com a base de dados "Eating & Health Module Dataset", que contém informações sobre hábitos alimentares, mas não possui uma variável de resposta (y). Este modelo ajudará a categorizar os usuários com base em padrões de alimentação saudável ou não, permitindo ajudar a identificar potenciais usuários mesmo sem uma classificação explícita.

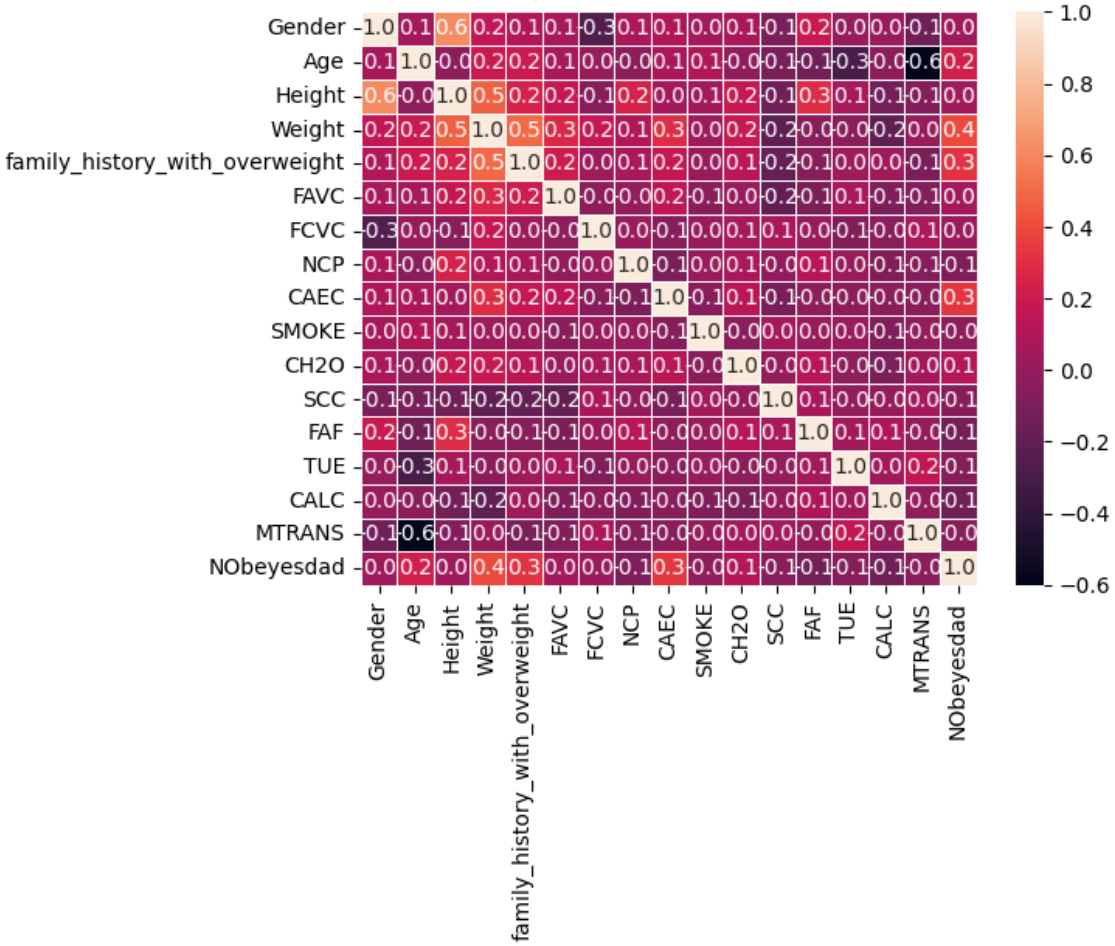
Definição dos modelos mais adequados para analisar os dados

A escolha do modelo foi principalmente baseada nos aprendizados em sala de aula, logo, não quis fugir muito do que nos foi passado. Por isso, escolhi estes três algoritmos para realizar as análises:

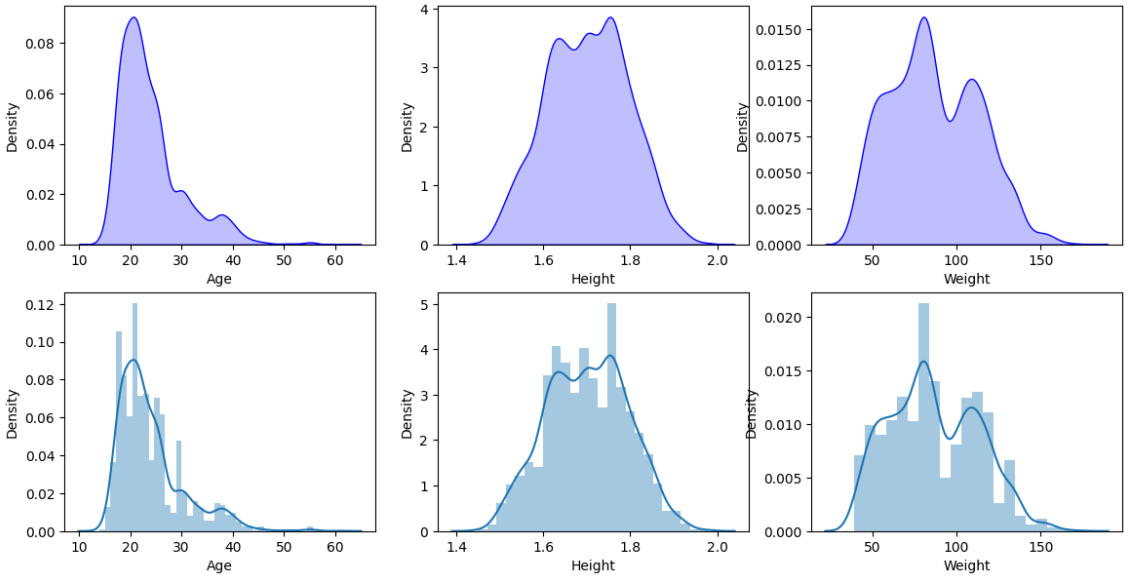
- Naive Bayes
- Decision Tree
- KNN

Estes foram os três modelos de classificação que nos foram apresentados, e serão os testados. As análises para entender as correlações dos meus dados com os algoritmos foram realizadas em detalhes nos arquivos que possuem os nomes dos modelos .ipynb, na pasta "analise_de_dados", no tópico "Entendendo o Resultado do Modelo". No entanto, irei destacar aqui o conteúdo principal que justifica a escolha desses modelos para teste.

Modelo Naive Bayes:

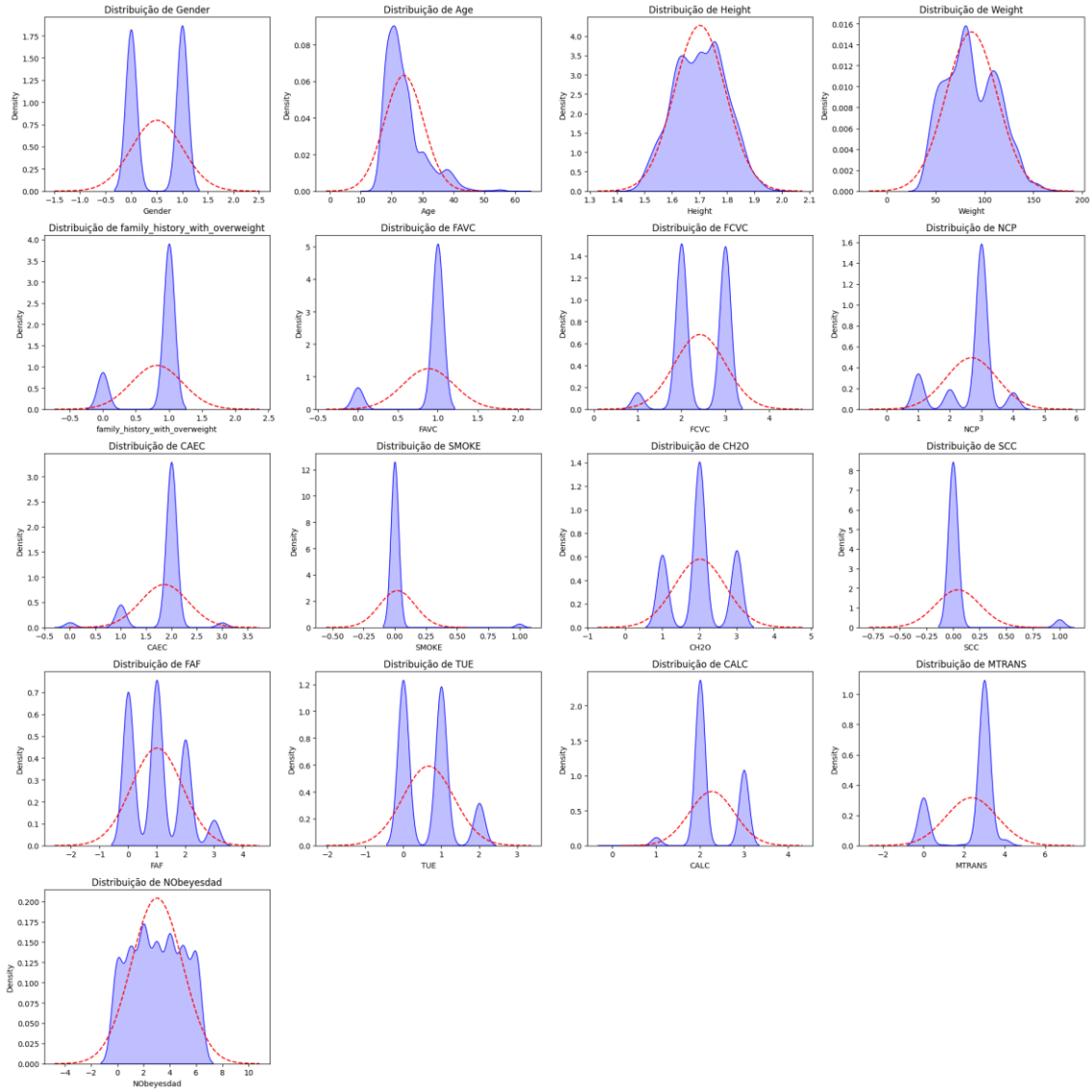


Ao plotarmos uma matriz de correlação, é possível observar que as variáveis são independentes, por apresentarem pouquíssima correlação entre si (números muito próximos a 0). Esse é um ponto positivo para o modelo, pois o algoritmo Naive Bayes é um classificador probabilístico que assume que as características (features) são independentes entre si, daí o termo "naive" (ingênuo). Essa é uma simplificação feita para facilitar o cálculo das probabilidades condicionais necessárias para a classificação.



O modelo "GaussianNB" assume que os dados quantitativos seguem uma distribuição normal (gaussiana). Ao analisarmos os gráficos plotados (distribuição dos dados quantitativos), fica claro que as colunas "Age" e "Weight" estão bem distantes de uma distribuição normal. Por outro lado, a coluna "Height" se aproxima bastante da distribuição normal, podendo talvez ter grande importância na predição.

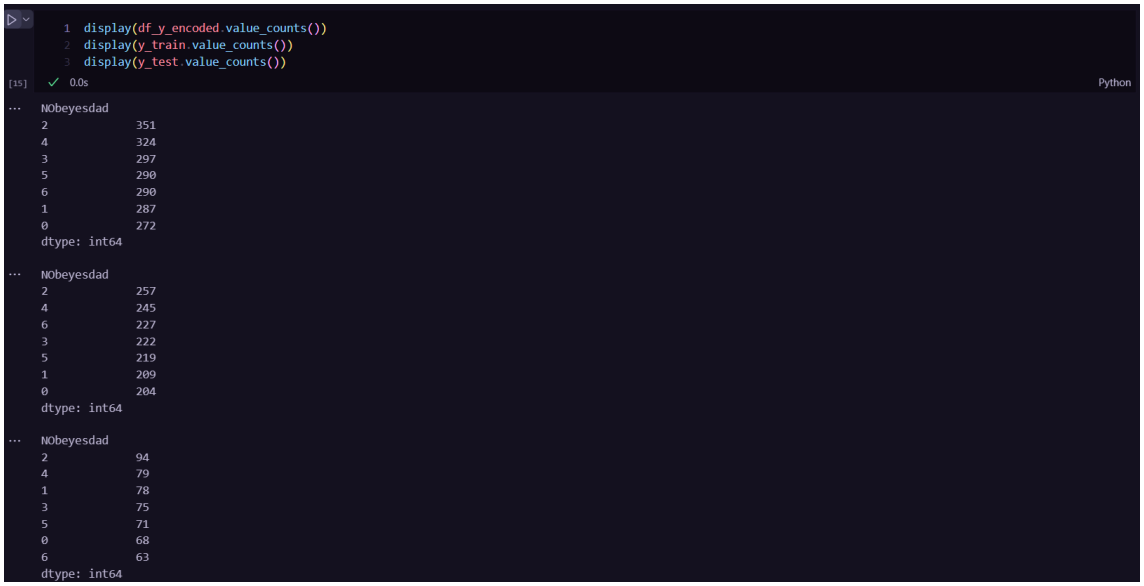
Modelo Decision Tree:



O modelo Decision Tree trabalha bem com características descritivas e bem distribuídas, além de classes balanceadas.

Podemos ver que possuímos muitas colunas, o que traz características suficientes para descrever adequadamente as classes de resposta. Os dados estão bem distribuídos, mas as classes descritivas não estão balanceadas.

Modelo KNN:



O bom desempenho do algoritmo KNN está fortemente ligado ao balanceamento das classes no conjunto de dados. Quando as classes estão equilibradas, o KNN consegue encontrar vizinhos de todas as classes de maneira mais justa, o que leva a uma classificação mais precisa.

Adendo:

Confiar unicamente na matemática para a escolha de modelos é uma abordagem limitada e potencialmente enganosa. Embora a matemática forneça fundamentos importantes e ajude a construir a base teórica dos modelos, é crucial lembrar que cada base de dados possui suas próprias características, peculiaridades e desafios específicos. A verdadeira eficácia de um modelo só pode ser verificada por meio de experimentação prática, testes rigorosos e análise empírica dos resultados. Sem a validação adequada através dos dados reais, qualquer confiança em um modelo permanece uma aposta, e não uma certeza.

Descrição dos modelos selecionados

É interessante ressaltar novamente que todos os modelos foram escolhidos por terem sido apresentados em sala de aula e por conta de sua simplicidade de aplicação. Mesmo assim, é importante conhecer seus aspectos principais. Esses aspectos podem nos ajudar a entender os resultados das IAs.

Aspectos Positivos do Bayes:

- Interpretação Fácil dos Resultados:

Robustez contra Overfitting: Como o modelo é simples e não tenta capturar interações complexas entre variáveis, ele é menos propenso ao overfitting em comparação com modelos mais sofisticados, como árvores de decisão complexas.

- Modelo Probabilístico Natural:

Como um modelo probabilístico, ele é capaz de lidar de forma intuitiva com problemas de classificação onde as variáveis de entrada estão em formatos categóricos, como classes ou rótulos.

Aspectos Positivos do KNN:

- Bom Desempenho em Conjuntos de Dados Pequenos:

Em um conjunto de dados pequeno, o KNN pode ter uma boa performance porque não precisa "aprender" padrões complexos. A classificação é feita com base nas instâncias já conhecidas, que são poucas e, portanto, fáceis de comparar e computar.

Além disso, como o KNN é sensível às distâncias entre os dados, se o conjunto de dados for pequeno, é mais provável que os dados estejam mais próximos uns dos outros e que as fronteiras de decisão entre classes sejam claras, tornando o KNN eficiente.

- Capacidade de Classificação Não Linear:

Em situações em que há uma fronteira de decisão complexa entre classes, o KNN cria regiões de decisão que podem ser muito irregulares, contornando esses padrões. Isso é particularmente útil em problemas onde os dados estão distribuídos de forma dispersa ou agrupada.

Aspectos Positivos do Decision Tree:

- Capacidade de Lidar com Dados Categóricos e Numéricos:

As árvores de decisão podem trabalhar diretamente tanto com dados categóricos quanto numéricos porque suas decisões são baseadas em divisões recursivas que buscam maximizar a separação entre classes ou minimizar a variância nos dados.

Por exemplo, se temos uma feature categórica x com valores "A", "B" e "C", a árvore pode dividir os dados dizendo: "Se x = 'A', vá para a esquerda; caso contrário, vá para a direita".

- Sem Necessidade de Suposições Sobre a Distribuição dos Dados:

Diferente de outros modelos estatísticos, como a Regressão Logística, que assume uma relação linear entre as variáveis explicativas e a probabilidade de cada classe, ou o Naive Bayes, que assume a independência condicional entre as features, as árvores de decisão não fazem nenhuma suposição específica sobre a forma ou distribuição dos dados.

Árvores de decisão não requerem pressupostos sobre:

- A distribuição gaussiana (normal) dos dados.

- A linearidade das relações entre as variáveis de entrada e saída.

- A homogeneidade da variância ou independência entre as variáveis.

Aplicação dos modelos selecionados

Todos os processos de criação das IAs utilizaram 75% dos dados para treino e 25% dos dados para teste, aplicando a semente 42. A escolha dessa semente se deve à sua popularidade e consistência no treinamento de modelos de machine learning, sendo frequentemente utilizada como referência para a reprodutibilidade dos resultados, conforme discutido no artigo "The Story Behind Random Seed 42 in Machine Learning". O uso da semente 42 assegura que, ao executar o código novamente com os mesmos dados e configurações, os resultados serão consistentes, uma prática comum na área para garantir a confiabilidade das comparações entre modelos. Além disso, no popular romance de ficção científica de Douglas Adams de 1979 , O Guia do Mochileiro das Galáxias , perto do final do livro, o supercomputador Deep Thought revela que a resposta para a grande questão da “vida, do universo e de tudo mais” é 42.

Artigo da seed 42 -> https://medium.com/geekculture/the-story-behind-random-seed-42-in-machine-learning-b838c4ac290a

No teste inicial dos modelos, não foram aplicados hiperparâmetros que não fossem obrigatórios. Isso significa que as configurações padrão dos algoritmos foram mantidas para uma primeira análise, sem ajustes manuais em parâmetros como o número de vizinhos no KNN, por exemplo. O objetivo dessa abordagem inicial é observar o desempenho básico dos modelos antes de proceder com otimizações mais detalhadas. Além disso, para verificar o risco de overfitting, foi realizada a validação cruzada (cross-validation) com 5 folds (partes), garantindo que cada partição dos dados respeitasse a mesma proporção de treino e teste que será utilizada na aplicação final do modelo.

Em todo o processo de avaliação das IAs, os principais fatores analisados foram a acurácia, o F1-score e a matriz de confusão. A acurácia mede a proporção de previsões corretas em relação ao total de previsões realizadas, sendo uma métrica geral de desempenho. No entanto, em casos de dados desbalanceados, onde uma classe pode estar mais representada do que outra, o F1-score é utilizado como métrica complementar, já que leva em consideração tanto a precisão quanto o recall, equilibrando a análise do desempenho do modelo em cenários de classes desiguais.

Além disso, a matriz de confusão foi gerada para uma análise gráfica das previsões. Essa matriz ajuda a visualizar como as previsões estão distribuídas em cada classe, permitindo identificar se o modelo está cometendo mais erros de falso positivo ou falso negativo, e em quais classes o modelo pode estar tendo mais dificuldades.

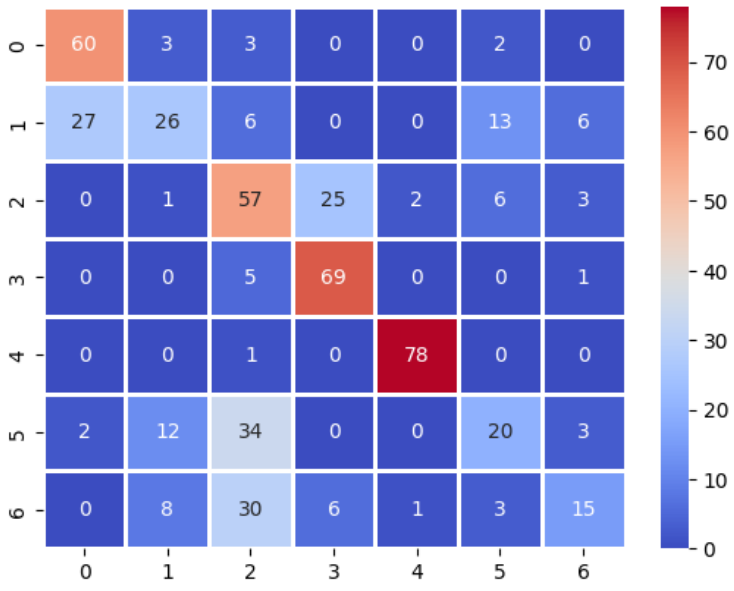
Em todo o processo de avaliação das IAs, os principais fatores analisados foram a acurácia, o f1_score e a matriz de confusão, para uma análise gráfica das previsões, ajudando a visualizar como as previsões estão distribuídas em cada classe.

Análise dos Resultados

A base de dados utilizada contém 16 perguntas, o que pode impactar a experiência do usuário. O teste inicial foi realizado com todas as colunas, mas nas próximas etapas, cada algoritmo será otimizado com ajustes de hiperparâmetros. O objetivo é reduzir o número de perguntas ao usuário, mantendo a melhor performance possível.

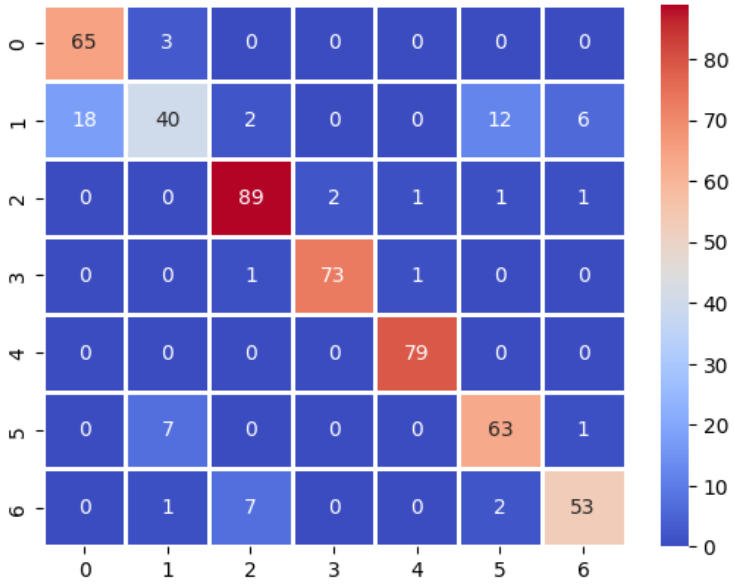
É interessante acrescentar que, ao analisarmos todos os modelos aplicados no Cross Validation, todos apresentaram ótimos resultados, descartando, em parte, overfitting e underfitting.

Bayes:



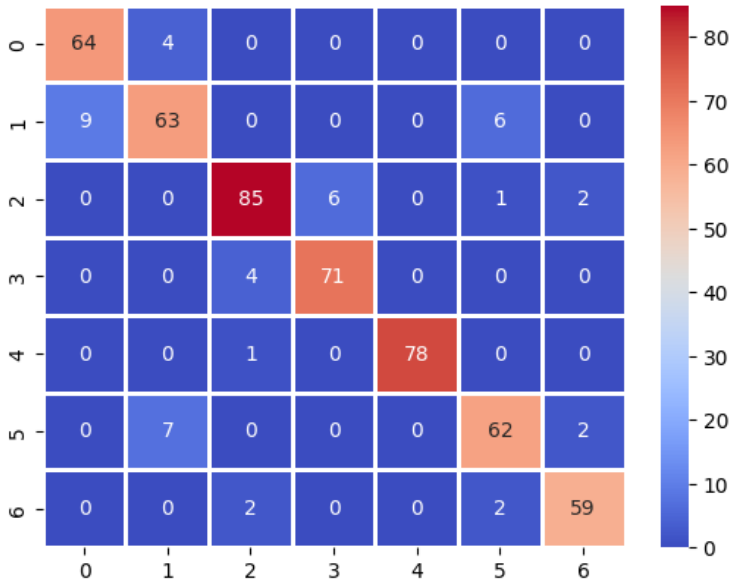
O algoritmo apresentou uma acurácia de 62%. A classe 4 teve o melhor desempenho, com 96% de precisão e 99% de recall, enquanto a classe 6 teve resultados mais modestos, com 54% de precisão e 24% de recall.

KNN:



O algoritmo obteve uma acurácia de 88%. As classes 2, 3 e 4 apresentaram os melhores resultados, com precisão e recall elevados. Outras classes tiveram desempenho ligeiramente inferior.

Decision Tree:



Com uma acurácia de 91%, o algoritmo apresentou resultados sólidos em todas as classes, com precisão e recall acima de 80%, destacando-se pela consistência nas métricas.

Comparação Entre Modelos:

Comparando os testes dos modelos, o **Decision Tree** apresentou o melhor desempenho geral, com acurácia de 91%, destacando-se em todas as classes, especialmente na classe 4, com f1-score de 0.99. O **KNN** também teve bom desempenho, com acurácia de 88%, sendo forte nas classes 3 e 4. Já o **Bayes** foi o que teve o pior desempenho, com acurácia de 62%, apresentando dificuldades nas classes 1 e 6, onde os f1-scores foram mais baixos. A classe 4 foi consistentemente a melhor classificada em todos os modelos.

Aplicação Aprofundada dos modelos selecionados

Após a primeira aplicação dos modelos, realizaremos uma segunda aplicação com o objetivo de otimizar ao máximo o desempenho de todos os modelos alinhados ao nosso propósito. Para isso, utilizei algumas tecnologias, como o **StandardScaler**, **SelectKBest**, **PCA** e a **Pipeline** do **sklearn** para orquestrar todo o processo.

De maneira geral, o processo pode ser descrito da seguinte forma:

- **"df_metricas"** → DataFrame que armazena o nome do modelo, o DataFrame utilizado, as métricas de teste e a melhor combinação de hiperparâmetros encontrada.
- **"pipe"** → Variável que contém todos os processos que compõem a pipeline.
    - StandardScaler → Centraliza os dados em torno de zero, subtraindo a média de cada coluna e dividindo pelo desvio padrão, normalizando a dispersão dos dados.
    - SelectKBest → Seleciona as features mais importantes da base de dados, auxiliando na redução de dimensionalidade. Utilizo esse método para reduzir o número de perguntas que serão feitas ao usuário, de 16 para, no máximo, 10.
    - PCA → Reduz a dimensionalidade após o SelectKBest, preservando o máximo de variância nos dados.
    - KNeighborsClassifier → Modelo de machine learning utilizado após a redução de dimensionalidade.
- **"params_pipe"** → Lista de parâmetros que serão testados na pipeline.
    - SelectKBest__k e pca__n_components → Definem o número de colunas a serem mantidas na redução dimensional.
    - SelectKBest__score_func → Funções de avaliação usadas para selecionar as colunas mais relevantes.
- **"valores_k"** → Define o número máximo de colunas a serem selecionadas (até 10) pelo SelectKBest.
- **GridSearchCV** → Executa a pipeline com diferentes combinações de hiperparâmetros, realiza validação cruzada e encontra a combinação que otimiza a acurácia.

Além disso, foram incluídas variações de hiperparâmetros para cada modelo:

Bayes:

- **model__var_smoothing** → Adiciona uma pequena quantidade à variância de cada feature, evitando problemas numéricos e overfitting.

KNN:

- **model__algorithm** → Especifica o algoritmo a ser usado para computar os vizinhos mais próximos.
- **model__metric** → Especifica a métrica de distância usada para calcular a proximidade entre os pontos.
- **model__weights** → Determina a forma como os vizinhos influenciam a predição.
- **model__n_neighbors** → Define o número de vizinhos a serem considerados para realizar a predição.

DecisionTree:

- **model__criterion** → Define a função de medida da impureza ou do ganho de informação durante o processo de construção da árvore. Controla como a árvore de decisão decide os melhores splits (divisões) nos dados, ou seja, como ela seleciona as variáveis e os valores nos nós que irão dividir os dados da maneira mais informativa.
- **model__ccp_alpha** → Representa o parâmetro de *Complexity Pruning Minimal Cost-Complexity* (Poda de Complexidade Mínima). É usado para controlar o processo de poda da árvore, visando reduzir o overfitting removendo ramos que têm pouca importância.
- **model__splitter** → Determina a estratégia usada para escolher a divisão em cada nó, podendo avaliar todos os atributos e todos os possíveis pontos de corte para encontrar a melhor divisão ou avaliar uma seleção aleatória de atributos em cada nó.

Análise dos Novos Resultados

Para analisarmos os resultados dos modelos, foram plotadas matrizes de confusão para cada base de dados treinada e testada. Além disso, os melhores resultados também foram armazenados no **df_metricas**, conforme explicado anteriormente.

A seguir, apresentamos alguns dos resultados obtidos para cada modelo:

Bayes:

À primeira vista, a mudança pode parecer pequena. No entanto, foi necessário limitar o modelo a no máximo 10 colunas de características, uma redução significativa em relação às 16 colunas utilizadas inicialmente no treinamento. Mesmo assim, o modelo conseguiu manter resultados semelhantes no teste simples e, em alguns casos, até melhores.

Outras conclusões que podem ser tiradas são que as colunas selecionadas que obtiveram o melhor desempenho na validação cruzada foram:

- Age
- Height
- Weight
- family_history_with_overweight
- FAVC
- CAEC

- CH20

Por fim, as métricas obtidas pelo modelo não foram das melhores, indicando que este pode não ser o modelo mais adequado para essa base de dados específica.

KNN:

O modelo KNN apresentou uma melhora significativa comparada ao primeiro teste, exibindo resultados mais que satisfatórios, com precisão, recall e acurácia elevadas.

Além disso, as colunas selecionadas que obtiveram o melhor desempenho na validação cruzada foram apenas:

- Height
- Weight

Este resultado é peculiar, considerando que a base possui 16 colunas e o modelo alcançou os melhores resultados com apenas 2 colunas. Contudo, a escolha dessas colunas faz muito sentido, uma vez que a relação entre peso e altura (IMC) descreve bem casos de sobrepeso e desnutrição.

Adicionalmente, o modelo está apresentando este desempenho com apenas 2 vizinhos como parâmetro, um valor considerado extremo. Isso faz sentido para o nosso caso, já que valores baixos de vizinhos são sensíveis a outliers. Como decidi manter os outliers de peso e altura por serem descritivos para a predição, o modelo validou essa abordagem com bons resultados.

Conclusão: O resultado do modelo KNN está muito bom, conseguindo predizer bem todas as 7 classes de resposta.

Decision Tree:

O modelo de Decision Tree manteve um ótimo desempenho, com precisão, recall e acurácia altas.

As colunas selecionadas que obtiveram o melhor desempenho na validação cruzada foram apenas:

- Height
- Weight

Este resultado é peculiar, considerando que a base possui 16 colunas e o modelo alcançou os melhores resultados com apenas 2 colunas. Contudo, a escolha dessas colunas faz muito sentido, uma vez que a relação entre peso e altura (IMC) descreve bem casos de sobrepeso e desnutrição.

Além disso, observamos que todos os processos realizados com a base de dados resultaram em uma distribuição quase perfeitamente normal das duas colunas selecionadas, o que contribuiu significativamente para o bom desempenho do modelo, compensando provavelmente as outras características que foram removidas.

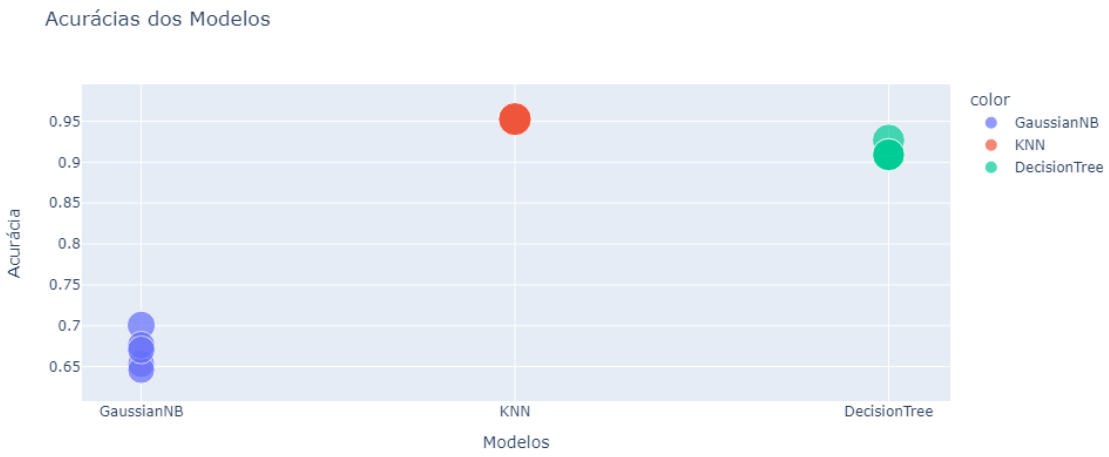
Conclusão: O resultado do modelo de Decision Tree está muito bom, conseguindo predizer bem todas as 7 classes de resposta.

Comparação dos Modelos

Para realizar as comparações entre os modelos, criei uma pasta específica **"interdisciplinar\supervisionado\comparacao_modelos"**, onde serão armazenados os resultados de cada modelo no **df_metricas**.

Primeiramente, para facilitar a análise das informações, plotei alguns gráficos com os resultados obtidos:

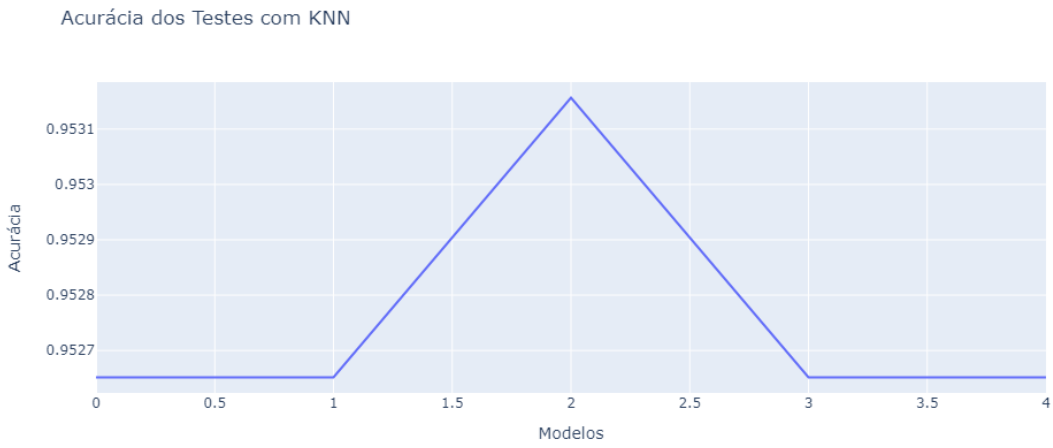
Desempenho dos Modelos:

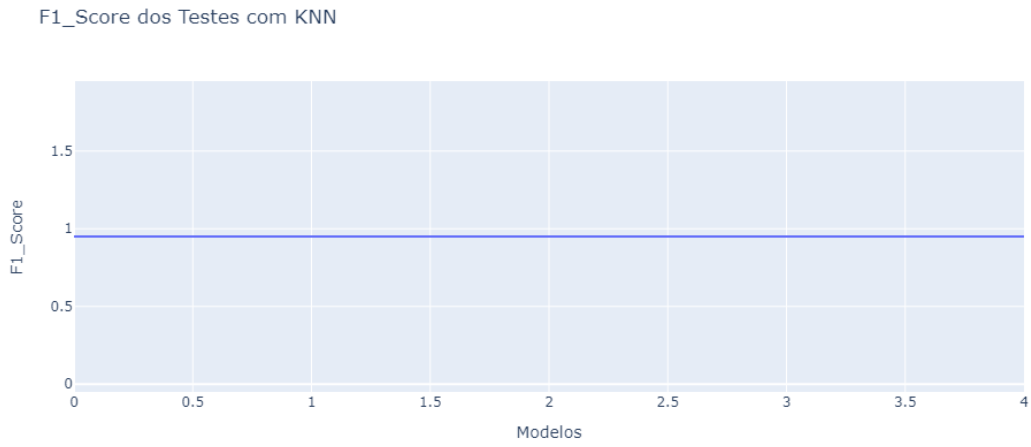


O modelo que apresentou o melhor desempenho nos critérios de **acurácia** e **F1_Score** foi o **KNN**.

Em seguida, serão realizadas outras análises para selecionar uma das variações testadas com o **KNN**.

Análise das Variações do Modelo KNN:





Os testes apresentaram valores muito semelhantes nas métricas, o que é esperado, já que a base de dados utilizada não afetou os resultados. Em todos os testes, foram escolhidas as mesmas duas colunas (**Height** e **Weight**).

Escolha dos Hiperparâmetros:

Dessa forma, a escolha será baseada nos hiperparâmetros. Ao analisar os hiperparâmetros, observamos que são praticamente iguais, com exceção de dois casos:

1. **Número de Vizinhos (n_neighbors)**:
    - Em um dos testes, o valor é **5**, diferente dos demais, que utilizam **2**.
2. **Métrica de Distância (metric)**:
    - Em outro teste, a métrica foi definida como **'euclidean'**, enquanto nos demais casos está definida como **'minkowski'**.

Considerando que a maioria dos testes utilizou **2** para n_neighbors e **'minkowski'** para metric, essas serão as opções escolhidas por serem as mais frequentes.

Decisão Final:

Portanto, optarei pelo **teste 0**, que atende a todos esses requisitos e utiliza a base de dados **"original"**.

Algoritmo Não Supervisionado

Todas as informações sobre o processo do algoritmo não supervisionado podem ser encontradas nos arquivos dentro da pasta "interdisciplinar\não_supervisionado*", mas apresentarei aqui um breve resumo de tudo que foi feito.

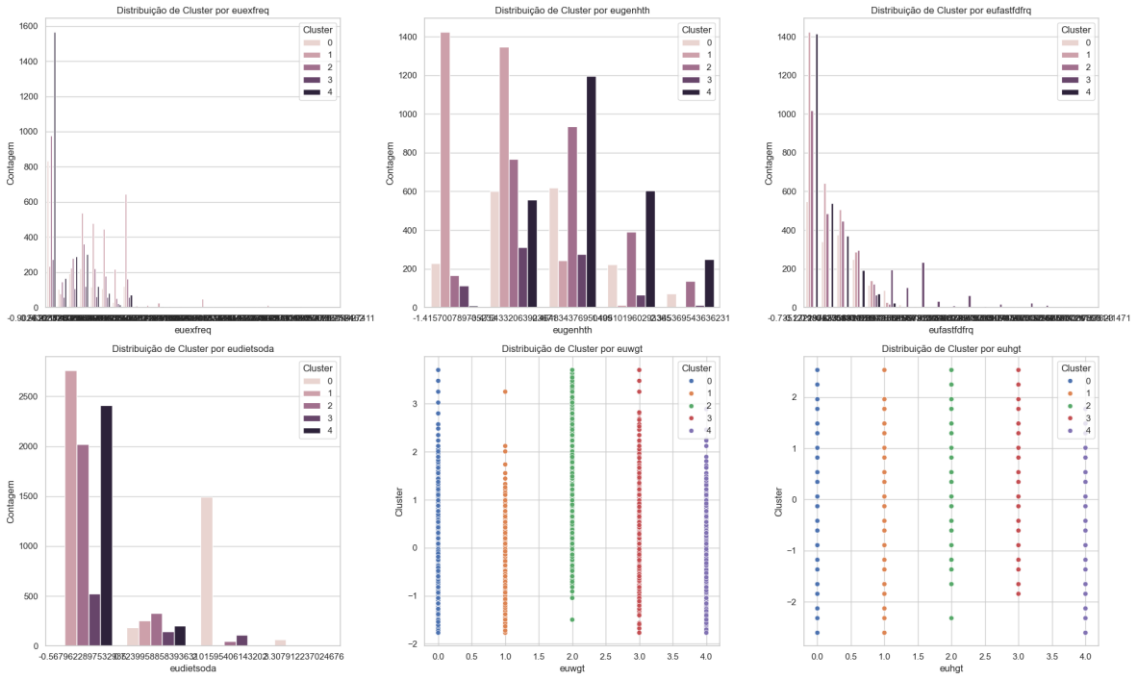
Primeiramente, comecei com uma análise exploratória da base de dados, selecionando as colunas que considerei mais relevantes sobre a rotina das pessoas. Após selecionar as colunas, comparei os valores da base com o dicionário de dados. Foi possível perceber alguns dados que não estavam presentes nas descrições do dicionário; como os valores eram poucos, optei por deletá-los ou arredondá-los para o valor válido mais próximo. Por fim, verifiquei a distribuição dos dados para identificar possíveis anomalias, mas não foram encontradas (apesar de haver outliers, eles são importantes para capturar rotinas extremas e, portanto, são descritivos).

Após toda a análise exploratória, comecei a aplicar os modelos de clusterização na base de dados. Fiz este processo em dois arquivos diferentes: um com a base normalizada e outro não. Em ambos, foram aplicados os seguintes modelos:

- K-means
- DBSCAN
- Agglomerative
- Mean Shift

Para comparar todos os modelos, criei uma pontuação que combina várias métricas existentes. Utilizei os scores "Coeficiente de Silhueta", "Índice de Davies-Bouldin" e "Índice de Calinski-Harabasz". Obtive os resultados de todos e calculei uma média, o que me permitiu ter uma visão geral da qualidade dos clusters, sem depender de uma métrica específica. É importante mencionar que não podia utilizar cada métrica separadamente, pois elas foram aplicadas como parâmetros para identificar os melhores hiperparâmetros.

Após selecionar a melhor clusterização de cada modelo com base na nossa pontuação criada, realizei uma análise dos clusters. Não podemos nos basear apenas na pontuação, pois ela nos retorna apenas a qualidade dos clusters e não o embasamento para a escolha deles. Assim, plotei a distribuição dos dados em cada cluster para ver qual possuía a melhor lógica de agrupamento. Segundo os critérios do projeto, o melhor modelo identificado foi o K-means com a base normalizada. Veja a seguir o gráfico da distribuição:



Após identificar o melhor modelo, comecei a caracterizar cada cluster com base nas análises realizadas. Estas foram as nossas caracterizações:

1. Cluster 1:

- **Frequência de Exercícios (EUEXFREQ):** Maior frequência de exercícios, o que é positivo para a saúde.
- **Autoavaliação da Saúde (EUGENHTH):** Melhor nota, indicando uma boa percepção da saúde física.
- **Consumo de Fast Food (EUFASTFDFRQ):** Consumo mediano, equilibrado.
- **Tipo de Refrigerante (EUDIETSODA):** Melhores valores, mais próximos de 0 (menor consumo de refrigerante).
- **Peso (EUWGT) e IMC:** Peso médio e IMC menor, sugerindo uma composição corporal saudável.
- **Altura (EUHGT):** Média, sem impacto negativo significativo na saúde.

Conclusão: Este cluster apresenta os melhores indicadores de saúde geral, combinando alta frequência de exercícios, boa autoavaliação da saúde, baixo consumo de refrigerantes e um IMC saudável.

2. Cluster 4:

- **Frequência de Exercícios (EUEXFREQ):** Menor frequência de exercícios, o que pode impactar negativamente a saúde.
- **Autoavaliação da Saúde (EUGENHTH):** Pior nota, indicando possíveis problemas de saúde ou percepção negativa.
- **Consumo de Fast Food (EUFASTFDFRQ):** Menor consumo semanal, o que é positivo.
- **Tipo de Refrigerante (EUDIETSODA):** Melhores valores, mais próximos de 0.
- **Peso (EUWGT) e IMC:** Pesos baixos e IMC médio, que podem ser positivos, mas devem ser avaliados em conjunto com outros fatores.

- **Altura (EUHGT):** Pessoas mais baixas, o que por si só não impacta negativamente a saúde.

Conclusão: Embora haja aspectos positivos como baixo consumo de fast food e refrigerantes, a menor frequência de exercícios e a pior autoavaliação da saúde colocam este cluster na segunda posição.

3. Cluster 2:

- **Frequência de Exercícios (EUEXFREQ):** Mediana, equilibrada.
- **Autoavaliação da Saúde (EUGENHTH):** Mediana.
- **Consumo de Fast Food (EUFASTFDFRQ):** Mediano.
- **Tipo de Refrigerante (EUDIETSODA):** Apenas mediano.
- **Peso (EUWGT) e IMC:** Maiores pesos e IMC elevado, o que pode indicar sobrepeso ou obesidade.
- **Altura (EUHGT):** Pessoas mais altas, geralmente associadas a melhores índices de saúde, mas o alto IMC contrabalança.

Conclusão: O alto peso e o IMC elevado são preocupantes para a saúde, colocando este cluster em uma posição intermediária, mas inferior aos anteriores devido aos riscos associados ao sobrepeso.

4. Cluster 3:

- **Frequência de Exercícios (EUEXFREQ):** Mediana.
- **Autoavaliação da Saúde (EUGENHTH):** Mediana.
- **Consumo de Fast Food (EUFASTFDFRQ):** Maior consumo, o que é negativo para a saúde.
- **Tipo de Refrigerante (EUDIETSODA):** Mediano com variações aos extremos.
- **Peso (EUWGT) e IMC:** Peso e IMC médios.
- **Altura (EUHGT):** Segundo lugar em pessoas mais altas.

Conclusão: O alto consumo de fast food é um fator de risco significativo, o que coloca este cluster abaixo dos anteriores em termos de saúde geral.

5. Cluster 0:

- **Frequência de Exercícios (EUEXFREQ):** Segunda menor frequência de exercícios.
- **Autoavaliação da Saúde (EUGENHTH):** Mediana.
- **Consumo de Fast Food (EUFASTFDFRQ):** Um pouco acima da média.
- **Tipo de Refrigerante (EUDIETSODA):** Piores valores, com alto consumo de refrigerantes.
- **Peso (EUWGT) e IMC:** Menores pesos e IMC médio, o que pode ser positivo, mas o alto consumo de refrigerantes e baixa frequência de exercícios comprometem a saúde.
- **Altura (EUHGT):** Média com variações extremas.

Conclusão: Este cluster apresenta os piores indicadores de saúde devido ao alto consumo de refrigerantes e baixa frequência de exercícios, mesmo com pesos mais baixos.

Por fim, utilizei a clusterização como classe Y para construir um modelo supervisionado. Isso permite utilizar a lógica dos clusters para classificar as pessoas. Todos os modelos testados

apresentaram um bom desempenho, todos com mais de 90% de acurácia e F1-score, mas o melhor foi o KNN. Com isso, precisei apenas serializar o modelo e utilizar toda a lógica para atribuir estes resultados com a outra IA criada.

Conclusão

Após ajustes iniciais, observamos que, embora os modelos aplicados se adequassem bem à base de predição de obesidade, as informações dessa base isoladamente não eram suficientes para identificar nossos potenciais usuários de maneira precisa. Para contornar essa limitação, introduzimos um modelo não supervisionado, que trouxe insights mais detalhados sobre os hábitos das pessoas, enriquecendo significativamente nossas predições. Com isso, conseguimos desenvolver um modelo de predição mais alinhado às necessidades do aplicativo.

Esse sistema agora nos permite identificar possíveis usuários que possam se beneficiar das principais funcionalidades do aplicativo, como a inclusão alimentar, o suporte psicológico por meio da nutrição e a personalização de dietas. Dessa forma, o aplicativo se torna mais eficaz em atender pessoas que buscam uma vida mais saudável e equilibrada, proporcionando uma experiência ajustada aos objetivos de bem-estar dos usuários e fortalecendo a conexão entre eles e o serviço oferecido.

RPA

Introdução

O RPA visa automatizar a atualização de dados cadastrais entre o banco de origem e o banco de dados normalizado e o MongoDB do aplicativo Let's Snack, assegurando que todas as informações estejam sincronizadas corretamente.

Funcionalidades

- Atualizar os dados cadastrais do banco origem para o banco de dados normalizado da 2° Série;
- Atualizar os dados cadastrais do banco origem para o banco de dados MongoDB do APP Let's Snack.

Tabelas Contempladas

A seguir, estão as tabelas que tiveram os dados transferidos.
Banco Origem -> Banco Destino

- Recipe -> let_recipes e let_preparation_methods
- Ingredient -> let_ingredients
- Restriction -> let_restrictions
- Admin -> let_adm
- Ingredient_Recipe -> let_recipes_ingredients e let_medition_types
- Recipe_restriction -> let_recipes_broken_restrictions

- Ingredient_restriction -> let_ingredients_broken_restrictions

Normalização
Para as tabelas Origem que se transformaram em duas tabelas no destino, foi aplicado o conceito de normalização

Recipe
- O campo ‘steps’ é uma string que representa uma lista, onde cada passo termina com ponto e vírgula (;), então foi criada uma tabela contendo o ID da receita, o passo e o número dele.

Ingredient_Recipe
- O campo ‘measure’ armazena as informações da medida do ingrediente utilizado na receita, e nele aplicamos a normalização criando uma tabela com os tipos de medições.

Publicação da IA no Aplicativo
Introdução
O objetivo é consumir um modelo de Machine Learning para prever se um usuário é um potencial usuário do aplicativo, ou seja, se ele se alinha ao perfil do público-alvo.

Aplicação
A classificação estará disponível na tela de cadastro do aplicativo para induzir o usuário a realmente utilizar o aplicativo.
Quando o usuário abrir a aplicação da IA, terá um formulário com as seguintes perguntas:
- Email
- Peso
- Altura
- Frequência de atividade físicas nos últimos 7 dias (de 1 a 40)
- Autoavaliação da saúde física de 1 a 5 (sendo 1 excelente e 5 ruim)
- Frequência de compra de fast food nos últimos 7 dias (1 a 30)
- Qual tipo de refrigerante você consume?
    - 0 – Não Consome
    - 1 – Diet
    - 2 – Normal
    - 3 – Ambos

Depois de responder as perguntas o app retornará um resultado falando se ele se alinha ou não ao perfil do público-alvo. E para isso, foi atribuído este método para classificar o usuário baseados nos resultados das IAs:

Classificações de Obesidade

Os pesos atribuídos às classificações de obesidade refletem a compatibilidade de cada grupo com os objetivos de saúde promovidos pelo aplicativo:

- **Peso Insuficiente (peso 2):** Embora abaixo do recomendado, este grupo pode estar mais inclinado a buscar melhorias de saúde, mostrando potencial alinhamento com o aplicativo.
- **Peso Normal (peso 3):** Representa o estado ideal de saúde, sendo altamente compatível com o público-alvo.
- **Sobrepeso Nível I (peso 1):** Caracteriza um leve sobrepeso, ainda próximo aos padrões de saúde desejados.
- **Sobrepeso Nível II (peso -1):** Um sobrepeso mais acentuado, que sugere um menor alinhamento com os objetivos do app.
- **Obesidade Tipos I, II e III (pesos -2 a -4):** Representam níveis crescentes de obesidade, associados a maiores riscos à saúde e uma menor probabilidade de adesão ao perfil do aplicativo.

Clusters de Hábitos

Para os clusters de hábitos, os pesos refletem o estilo de vida e o grau de aderência aos propósitos do aplicativo:

- **Cluster 1 (peso 4):** Agrupa indivíduos com os hábitos mais saudáveis, como alta frequência de exercícios e consumo equilibrado de alimentos, representando os usuários mais alinhados.
- **Cluster 4 (peso 1):** Embora com algumas limitações, este grupo mantém hábitos que o tornam um potencial usuário do app.
- **Cluster 2 (peso 0):** Reflete um perfil moderado, com hábitos equilibrados, mas que pode se beneficiar de estímulos para melhorar a saúde.
- **Clusters 3 e 0 (peso -4):** Comportamentos de risco para a saúde, incluindo alto consumo de fast food e refrigerantes e baixa frequência de exercícios, indicando menor compatibilidade com a proposta do aplicativo.

Conclusão

A atribuição desses pesos possibilita identificar e priorizar usuários cujos perfis de saúde e hábitos de vida estão mais alinhados com o objetivo do nosso aplicativo. Dessa forma, os esforços são direcionados a um público com maior potencial de engajamento e que se mostra inclinado a adotar ou manter um estilo de vida saudável, promovendo uma experiência mais significativa e alinhada aos propósitos da plataforma.

Por fim, o resultado da pesquisa é inserido no banco de dados para que, em um plano futuro, possa ser gerado relatórios para verificar se a IA está prevendo corretamente potenciais usuários.