Análise de Fatores de Sucesso para Redução de Gordura Corporal

Equipe: Nina da Hora

Integrantes da Squad

Bruna de Avila Pospiesz
Francielle Cristina da Costa Silva
Gisela Keidel
Ingrid Costa Ferreira
Luana Jaime Tocchio
Tatiana Varona Villavicencio
Vanelle Rabelo do Nascimento
Vanessa Simão da Costa

Equipe: Nina da Hora

Situação Atual

Uma empresa, especializada em consultoria fitness e nutrição, busca aprimorar a eficácia dos planos oferecidos aos clientes. Atualmente, as recomendações fornecidas são amplas e pouco personalizadas, o que limita os resultados. Além disso, ainda não há clareza sobre quais combinações de dieta e treino são realmente mais eficientes para reduzir o percentual de gordura corporal.

Objetivo do Projeto

- Construir um pipeline ETL totalmente integrado.
- Estruturar os dados seguindo a arquitetura Medalhão (Bronze, Silver e Gold).
- Gerar uma camada Gold limpa, tratada e pronta para análise.
- Conectar a base final ao Power BI para criação dos dashboards.
- Responder às principais perguntas de negócio por meio de visualizações objetivas.
- Oferecer insights acionáveis para nutricionistas e treinadores, apoiando decisões estratégicas.

Base de Dados

A análise foi realizada com uma base contendo 20.000 registros relacionados a estilo de vida, tipos de treino, padrões alimentares e métricas corporais dos indivíduos.

- Fonte dos dados: Kaggle.

Foram selecionadas 9 variáveis essenciais, que permitem responder às principais perguntas do negócio:

- Fat Percentage (Percentual de Gordura)
- Weight (kg)
- BMI (IMC)
- diet_type (Tipo de Dieta)
- Workout_Type (Tipo de Treino)
- Workout_Frequency (Frequência de Treino)
- Age (Idade)
- Gender (Gênero)
- Experience_Level (Nível de Experiência)

Primeiros Passos: Entendendo os Dados

Explorando os Dados

```
# carregando o arquivo CSV
df = pd.read_csv("estilo_de_vida.csv")
```

```
# Lendo as primeiras 5 linhas do df
df.head()
```

	Age	Gender	Weight (kg)	Height (m)	Max_BPM	Avg_BPM	Resting_BPM	Session_Duration (hours)	Calories_Burned	Workout_Type
0	34.91	Male	65.27	1.62	188.58	157.65	69.05	1.00	1080.90	Strength
1	23.37	Female	56.41	1.55	179.43	131.75	73.18	1.37	1809.91	HIIT
2	33.20	Female	58.98	1.67	175.04	123.95	54.96	0.91	802.26	Cardio
3	38.69	Female	93.78	1.70	191.21	155.10	50.07	1.10	1450.79	HIIT
4	45.09	Male	52.42	1.88	193.58	152.88	70.84	1.08	1166.40	Strength

5 rows × 54 columns

Colunas da Tabela

```
# Mostrar todas as colunas (lista nomes)
df.columns
```

```
Index(['Age', 'Gender', 'Weight (kg)', 'Height (m)', 'Max_BPM', 'Avg_BPM',
       'Resting_BPM', 'Session_Duration (hours)', 'Calories_Burned',
       'Workout_Type', 'Fat_Percentage', 'Water_Intake (liters)',
       'Workout_Frequency (days/week)', 'Experience_Level', 'BMI',
       'Daily meals frequency', 'Physical exercise', 'Carbs', 'Proteins',
       'Fats', 'Calories', 'meal_name', 'meal_type', 'diet_type', 'sugar_g',
       'sodium_mg', 'cholesterol_mg', 'serving_size_g', 'cooking_method',
       'prep_time_min', 'cook_time_min', 'rating', 'Name of Exercise', 'Sets',
       'Reps', 'Benefit', 'Burns Calories (per 30 min)', 'Target Muscle Group',
       'Equipment Needed', 'Difficulty Level', 'Body Part', 'Type of Muscle',
       'Workout', 'BMI_calc', 'cal_from_macros', 'pct_carbs', 'protein_per_kg',
       'pct_HRR', 'pct_maxHR', 'cal_balance', 'lean_mass_kg', 'expected_burn',
       'Burns Calories (per 30 min)_bc', 'Burns_Calories_Bin'],
      dtype='object')
```

Objetivo do Desafio

Camada Bronze

Extração

```
# ============================
# EXTRAÇÃO DE DADOS
# ============================

def coletar_dados(estilo_de_vida_csv, bronze_path = 'estilo_de_vida_bronze.csv' ):
    print(Fore.CYAN + '\nEtapa 1: Extraindo dados...\n')
    df = pd.read_csv(estilo_de_vida_csv)
    print('Dados originais:')
    print(df, '\n')
    # salvando bronze
    df.to_csv(bronze_path, index = False)
    print(Fore.CYAN + f'Etapa 1: Arquivo salvo em: {os.path.abspath(bronze_path)}\n')
    return df
```

- Carregar o dataset estilo_de_vida.csv
- Explorar os dados: visualizar os dados originais
- Salvando como como camada Bronze

```
Dados originais:
        Age  Gender  Weight (kg)  Height (m)  Max_BPM  Avg_BPM  Resting_BPM  ...   pct_HRR  pct_maxHR  cal_balance  lean_mass_kg  expected_burn  Burns Calories (per 30 min)_bc  Burns_Calories_Bin
0     34.91    Male        65.27        1.62   188.58   157.65        69.05  ...  0.741237   0.835985       725.10      47.777394         685.1600                    7.260425e+19              Medium
1     23.37  Female        56.41        1.55   179.43   131.75        73.18  ...  0.551247   0.734270      -232.91      40.809803         978.6184                    1.020506e+20                High
2     33.20  Female        58.98        1.67   175.04   123.95        54.96  ...  0.574534   0.708124       805.74      44.635580         654.5266                    1.079607e+20                High
3     38.69  Female        93.78        1.70   191.21   155.10        50.07  ...  0.744155   0.811150      1206.21      63.007432         773.6300                    8.987921e+19                High
4     45.09    Male        52.42        1.88   193.58   152.88        70.84  ...  0.668405   0.789751       303.60      43.347504         711.4176                    5.264685e+19                 Low
...     ...     ...          ...         ...      ...      ...          ...  ...       ...        ...          ...           ...            ...                             ...                 ...
19995 46.77  Female        98.31        1.90   199.20   148.18        63.72  ...  0.623413   0.743876      1865.39      71.269345         533.2558                    7.924402e+19              Medium
19996 40.38  Female        88.12        1.87   196.18   134.18        54.04  ...  0.563810   0.683964       173.87      65.049689        1310.6016                    5.708474e+19                 Low
19997 50.31    Male        46.20        1.67   163.34   157.92        61.65  ...  0.946701   0.966818       -43.80      35.420708         957.9568                    9.101285e+19                High
19998 52.36    Male        44.30        1.62   179.27   121.23        60.88  ...  0.509756   0.676243       346.25      35.889260         928.4004                    5.246436e+19                 Low
19999 29.56    Male        58.63        1.61   198.07   121.74        72.05  ...  0.394302   0.614631       804.16      44.303237         841.1040                    8.742467e+19                High
```

```
linhas = df.shape[0]
colunas = df.shape[1]

print("Linhas:", linhas)
print("Colunas:", colunas)
```

```
Linhas: 20000
Colunas: 54
```

Camada Silver

Transformação

- Carregando o dataset estilo_de_vida_bronze.csv
- Escolhendo as 9 colunas a serem trabalhadas

```
# ============================
# TRANFORMAÇÃO DOS DADOS
# ============================

def transformar_dados(bronze_path, silver_path: 'estilo_de_vida_silver.csv'):
    # Escolhendo as colunas a serem trabalhadas
    print(Fore.CYAN + '\nEtapa 2: Transformando dados...\n')
    print(Fore.CYAN + '\nEscolhendo as colunas a serem trabalhadas...\n')
    df_tratado = pd.read_csv(bronze_path)
    df_tratado = df_tratado[['Gender', 'Age', 'Weight (kg)', 'Fat_Percentage', 'BMI', 'diet_type', 'Workout_Type',
                             'Workout_Frequency (days/week)', 'Experience_Level']]
```

	Gender	Age	Weight (kg)	Fat_Percentage	BMI	diet_type	Workout_Type	Workout_Frequency (days/week)	Experience_Level
0	Male	34.91	65.27	26.800377	24.87	Vegan	Strength	3.99	2.01
1	Female	23.37	56.41	27.655021	23.48	Vegetarian	HIIT	4.00	2.01
2	Female	33.20	58.98	24.320821	21.15	Paleo	Cardio	2.99	1.02
3	Female	38.69	93.78	32.813572	32.45	Paleo	HIIT	3.99	1.99
4	Male	45.09	52.42	17.307319	14.83	Vegan	Strength	4.00	2.00

Transformação

```
# Renomeando as colunas
print(Fore.CYAN + '\nRenomeando as colunas...\n')
df_tratado = df_tratado.rename(columns={
"Gender" : "genero",
"Age" : "idade",
"Weight (kg)" : "peso",
"Fat_Percentage" : "percentual_de_gordura",
"BMI" : "imc",
"diet_type" : "tipo_de_dieta",
 "Workout_Type" : "tipo_de_treino",
 "Workout_Frequency (days/week)" : "frequencia_de_treino_semanal",
"Experience_Level" : "nivel_de_experiencia"
})
print(df_tratado.head())
```

- Renomeando as colunas
- Visualizando as 5 primeiras linhas

	genero	idade	peso	percentual_de_gordura	imc	tipo_de_dieta	tipo_de_treino	frequencia_de_treino_semanal	nivel_de_experiencia
0	Male	34.91	65.27	26.800377	24.87	Vegan	Strength	3.99	2.01
1	Female	23.37	56.41	27.655021	23.48	Vegetarian	HIIT	4.00	2.01
2	Female	33.20	58.98	24.320821	21.15	Paleo	Cardio	2.99	1.02
3	Female	38.69	93.78	32.813572	32.45	Paleo	HIIT	3.99	1.99
4	Male	45.09	52.42	17.307319	14.83	Vegan	Strength	4.00	2.00

Transformação

```
# Criando listas
print(Fore.CYAN + '\nCriando listas...\n')
df_tratado["genero"].unique().tolist()
df_tratado["tipo_de_dieta"].unique().tolist()
df_tratado["tipo_de_treino"].unique().tolist()
df_tratado["nivel_de_experiencia"].unique().tolist()

# Traduzindo
print(Fore.CYAN + '\nTraduzindo para português...\n')
df_tratado["genero"] = df_tratado["genero"].map({
"Male": "Masculino",
"Female": "Feminino"
})
df_tratado["tipo_de_dieta"] = df_tratado["tipo_de_dieta"].map({
"Vegan": "Vegana",
"Vegetarian": "Vegetariana",
"Paleo": "Paleolítica",
"Keto": "Cetogênica",
"Low-Carb": "Baixo Carboidrato",
"Balanced": "Balanceada"
})
df_tratado["tipo_de_treino"] = df_tratado["tipo_de_treino"].map({
"Strength": "Força",
"HIIT": "HIIT",
"Cardio": "Cardio",
"Yoga": "Yoga"
})
```

- Criando as listas
- Traduzindo as informações dos dados

Transformação

```python
# Arredondamento para 1 Casa Decimal
print(Fore.CYAN + '\nArredondando número para uma casa decimal...\n')
df_tratado["percentual_de_gordura"] = df_tratado["percentual_de_gordura"].round(1)
df_tratado["peso"] = df_tratado["peso"].round(1)

# Transformando colunas para int
print(Fore.CYAN + '\nTransformando colunas para int...\n')
colunas = ['frequencia_de_treino_semanal', 'idade', 'nivel_de_experiencia']
df_tratado[colunas] = df_tratado[colunas].astype(int)


# Substituição dos Códigos de Experiência por Categorias
print(Fore.CYAN + '\nSubstituindo códigos de experiência por categorias...\n')
df_tratado["nivel_de_experiencia"] = df_tratado["nivel_de_experiencia"].map({
    1: "Iniciante",
    2: "Intermediário",
    3: "Avançado"
})
print(df_tratado.head())
# salvando a camada Silver
df_tratado.to_csv(silver_path, index = False)
print(Fore.CYAN + f'Etapa 2: Arquivo salvo em: {os.path.abspath(silver_path)}\n')
```

- Arredondando os valores para 1 casa decimal
- Transformando colunas para int
- Substituindo os Códigos de Experiência por Categorias
- Salvando a camada Silver

Camada Gold

Transformação

- Classificação do Índice de Massa Corporal

```
# ==============================
# ETAPA GOLD
# ==============================

    print(Fore.CYAN + f'Etapa Gold inciada\n')
    print(Fore.CYAN + f'Criando a função de classificação do índice de massa corporal\n')
    return df_tratado

# Função de Classificação do Índice de Massa Corporal
def classificar_imc(imc):
    if imc < 18.5:
        return "Abaixo do Peso"
    elif 18.5 <= imc <= 24.9:
        return "Peso Normal"
    elif 25.0 <= imc <= 29.9:
        return "Sobrepeso"
    elif 30.0 <= imc <= 34.9:
        return "Obesidade Grau I"
    elif 35.0 <= imc <= 39.9:
        return "Obesidade Grau II"
    else:
        return "Obesidade Grau III"
```

Transformação

```python
def gerar_gold(silver_path='estilo_de_vida_silver.csv', gold_path='estilo_de_vida_gold.csv'):
    # Lendo o arquivo silver
    df_gold = pd.read_csv(silver_path)

    # Aplica o classificador de forma segura
    df_gold["classificacao_imc"] = df_gold["imc"].apply(classificar_imc)

    # mover para a 6ª posição (índice 5)
    colunas = list(df_gold.columns)
    colunas.insert(5, colunas.pop(colunas.index("classificacao_imc")))
    df_gold = df_gold[colunas]

    # Salva GOLD
    df_gold.to_csv(gold_path, index=False)
    print(Fore.CYAN + f'Etapa Gold concluída e salva em: {os.path.abspath(gold_path)}\n')
    print(df_gold.head())
    return df_gold
```

- Lendo o dataset silver
- Aplicando o classificador de forma segura
- Reordenando as colunas
- Salvando a camada Gold

	genero	idade	peso	percentual_de_gordura	imc	classificacao_imc	tipo_de_dieta	tipo_de_treino	frequencia_de_treino_semanal	nivel_de_experiencia
0	Masculino	34	65.3	26.8	24.87	Peso Normal	Vegana	Força	3	Intermediário
1	Feminino	23	56.4	27.7	23.48	Peso Normal	Vegetariana	HIIT	4	Intermediário
2	Feminino	33	59.0	24.3	21.15	Peso Normal	Paleolítica	Cardio	2	Iniciante
3	Feminino	38	93.8	32.8	32.45	Obesidade Grau I	Paleolítica	HIIT	3	Iniciante
4	Masculino	45	52.4	17.3	14.83	Abaixo do Peso	Vegana	Força	4	Intermediário

Carga – Banco de Dados

- Criação do Banco de Dados

```
# ==============================
# CRIAÇÃO DO BANCO DE DADOS
# ==============================
def carregar_dados(df, df_tratado, df_gold):
    print(Fore.CYAN + '\nEtapa 3: Criação do Banco de Dados...\n')
    print( Fore.YELLOW + '\nCriando um banco SQLite (arquivo estilo_de_vida.db).....')
    # cria o banco
    engine = create_engine('sqlite:///estilo_de_vida.db')
    # salva todas as tabelas no banco criado
    df.to_sql('estilo_de_vida_bronze', con=engine, if_exists='replace', index=False)
    df_tratado.to_sql('estilo_de_vida_silver', con=engine, if_exists='replace', index=False)
    df_gold.to_sql('estilo_de_vida_gold', con=engine, if_exists='replace', index=False)
    print( Fore.YELLOW + '\nBanco de Dados criado.....')
```

Carga - Automação do Pipeline de Dados

```python
def pipeline_etl():
    print(Fore.CYAN + '\nETL iniciada...\n')
    estilo_de_vida_csv= 'estilo_de_vida.csv'
    bronze_path = 'estilo_de_vida_bronze.csv'
    silver_path = 'estilo_de_vida_silver.csv'
    gold_path = 'estilo_de_vida_gold.csv'

    # 1) Extração
    df = coletar_dados(estilo_de_vida_csv, bronze_path)
    # 2) Transformação (gera silver)
    df_tratado = transformar_dados(bronze_path, silver_path)
    # 3) GOLD
    df_gold = gerar_gold(silver_path, gold_path)
    # 4) Carregamento
    carregar_dados(df, df_tratado, df_gold)

pipeline_etl()
```

- Criação do Pipeline

Principais Gráficos do Dashboard

- Construímos um dashboard que permite analisar os dados e identificar padrões consistentes sobre os fatores que impactam a redução de gordura corporal.

Indicadores gerais:

- Média do percentual de gordura: 26,1%
- Média de IMC: 24,92
- Total de participantes analisados: 20.000

Insight 1: Gênero

- Homens (26,14%) e mulheres (26,07%) possuem médias muito próximas, indicando que o gênero não é um fator determinante.

Insight 2: Dietas variam pouco

- As dietas apresentam diferenças pequenas entre si. Vegetarianos (25,87%)e adeptos da Paleolítica (25,95%) tiveram resultados ligeiramente melhores, com Veganos (26,28%) com resultado um pouco maior.

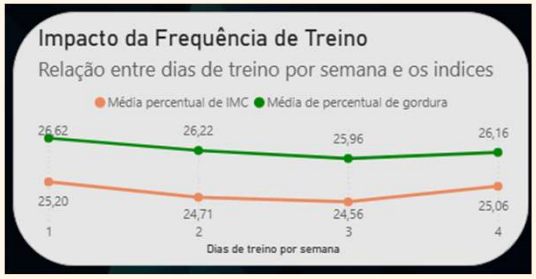
Insight 3: Tipo de treino varia pouco

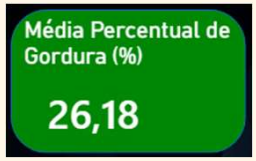
- Os tipos de treinos também tem valores muito próximos entre si, com uma diferença mais relevante quando analisado pelo gênero: Homens melhor desempenho com cardio (26,08%) e Mulheres com treinos de força (25,95%).

Insight 4: Frequência de treino – fator mais importante

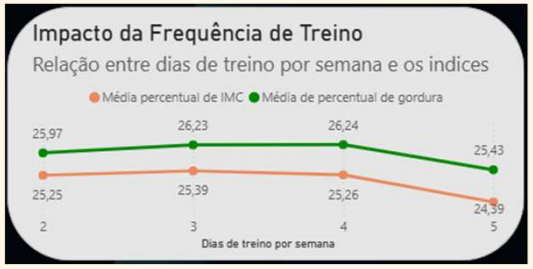
- O nível de experiência foi um dos fatores mais consistentes. Participantes avançados, que treinam de 3 a 5 vezes na semana, tem menor percentual de gordura, independente do gênero.

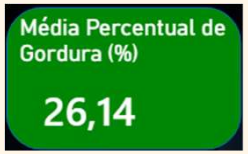
- Iniciante



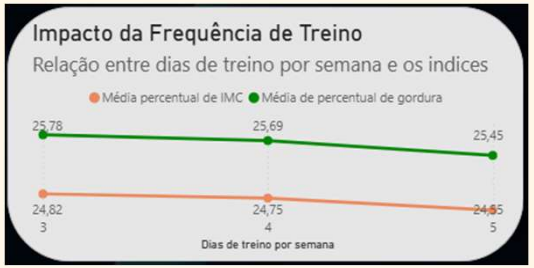


- Intermediário



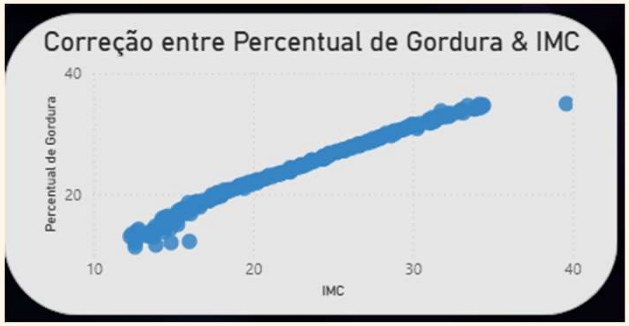


- Avançado

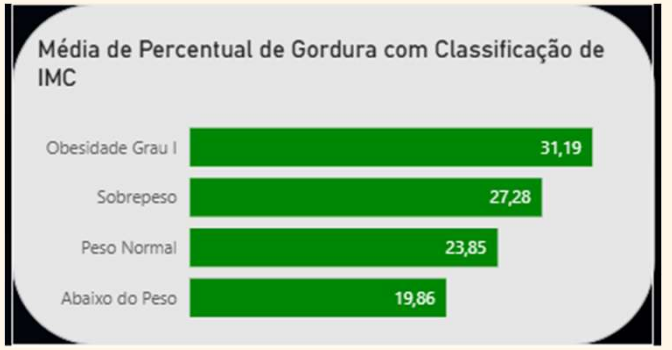


Insight 5: IMC x Gordura

- O gráfico confirma a forte correlação entre IMC e gordura corporal.



- Tipo de experiência avançado:



Conclusão

- Os tipos de dietas apresentaram pequena diferença entre elas.
- Os tipos de treino também mostram resultados próximos a média geral, com uma pequena diferença do treino de cardio para os homens e força para mulheres , que apresentaram melhores desempenhos.
- A grande descoberta é que a consistência e frequência importam mais do que a dieta ou modalidade isolada.

Recomendações

Recomendações práticas com base nos dados:

- Consistência do treino de 3 a 5 vezes na semana
- Progredir os níveis de experiência
- Variar os treinos
- Dietas sustentáveis
- Acompanhar pessoas com IMC acima da media
- Campanhas educativas

Próximos passos

- Adicionar novas variáveis
- Realizar análises mais profundas
- Evoluir dashboard
- Desenvolver versão mobile do dashboard

Organização da Equipe

Distribuição das atividades de acordo com os pontos fortes de cada integrante.

Comunicação através do Discord e Google Meet.

Facilidades e Dificuldades

Facilidades:

- Dedicação e colaboração do grupo: todas exploraram suas habilidades.
- Facilidade com a parte do Código (ETL): para algumas integrantes a etapa de extração e transformação foi mais tranquila.

Dificuldades:

- Erros e bugs inesperados das ferramentas: falhas no Power BI e no ambiente Python.
- Desafios com GitHub: dificuldades para clonar, versionar e enviar arquivos corretamente.
- Integração das partes do projeto: Sincronizar ETL, banco de dados e dashboard demandou mais atenção.

Agradecimento

Agradecemos a atenção de todas!

Acesse nosso repositório : escaneie o QR CODE