

**5º SEMESTRE – NOV/2025**  
**Desenvolvimento de Software Multiplataforma**  
**Fatec Franca - Dr Thomaz Novelino**

# **Análise de Aprendizagem de Máquina do banco de dados da PetDex**



O conjunto de scripts desenvolvidos tem como objetivo preparar, tratar e estruturar a base de dados da PetDex para o treinamento de um modelo de Inteligência Artificial capaz de **classificar possíveis doenças em animais com base em seus sintomas**.

As etapas descritas a seguir detalham o processo completo de limpeza, padronização, agrupamento e conversão dos dados, garantindo qualidade e consistência para a etapa de aprendizado de máquina.



Felipe Avelino Pedaes



Gabriel Resende Spirlandelli



Henrique Almeida Florentino



Luiz Felipe Vieira Soares



# 1. Tratamento Inicial do CSV: 01\_tratamento.py

Esta etapa foi responsável por normalizar e limpar os dados brutos extraídos da planilha original 00\_tabela.xlsx, gerando um novo arquivo tratado (01\_tabela\_tratada.xlsx).

## Principais ações realizadas:

### Remoção de acentos e caracteres especiais

Função `remover_acentos()` converte textos para o formato ASCII, eliminando acentuação (ex: “Fêmea” -> “Femea”). Isso garante compatibilidade com ferramentas e evita duplicações por grafias diferentes.

### Padronização de colunas

Todos os nomes de colunas foram convertidos para letras minúsculas e underscores (\_) substituíram espaços, mantendo um padrão técnico legível (ex: “Tipo de Doença” -> `tipo_de_doenca`).

### Conversão de respostas textuais em valores binários

“SIM” e “NÃO” foram substituídos por 1 e 0, tornando os dados compatíveis com algoritmos numéricos.



# 1. Tratamento Inicial do CSV: 01\_tratamento.py

## Principais ações realizadas:

### Padronização do gênero

“Macho” e “Fêmea” foram convertidos para 1 e 0 respectivamente.

### Conversão da duração em dias

Textos como “2 semanas” foram transformados em valores inteiros equivalentes (14 dias).

Isso facilita análises temporais e modelos que usam variáveis contínuas.

### Formatação uniforme de textos

Todas as strings foram colocadas em minúsculas e com underscores, padronizando a base.

### Saída final:

01\_tabela\_tratada.xlsx: base limpa e normalizada, pronta para análises estruturadas.



## 2. Expansão e Agrupamento de Sintomas: 02\_tratamento\_sintomas.py

O segundo script teve como objetivo organizar e consolidar as colunas de sintomas presentes na base, agrupando variações semelhantes e criando novas colunas booleanas para cada tipo de sintoma.

### Principais etapas:

#### Identificação automática de colunas de sintomas

O código busca todas as colunas com prefixo `sintoma_`.

#### Contagem e limpeza dos sintomas

Valores nulos e vazios foram eliminados, e os sintomas foram contados para identificar sua frequência.

#### Agrupamento manual de sinônimos e variações

Um dicionário (`agrupamentos_sintomas`) foi criado para reunir sintomas equivalentes sob uma mesma categoria. Exemplos:

“tosse seca”, “tosse forte” -> **tosse**

“agitacao leve”, “hiperatividade”, “latidos excessivos” -> **agitação**

“secrecao ocular clara”, “lacrimejamento” -> **secreção ocular**



## 2. Expansão e Agrupamento de Sintomas: 02\_tratamento\_sintomas.py

### Principais etapas:

#### Criação de novas colunas booleanas (0/1)

Cada grupo de sintomas tornou-se uma coluna própria, recebendo valor 1 quando presente no registro.

#### Remoção das colunas originais e salvamento dos resultados

O arquivo resultante (02\_tabela\_expandida.xlsx) contém apenas as colunas relevantes e já agrupadas.

### Saídas geradas:

**02\_tabela\_expandida.xlsx: base expandida e padronizada.**

**02\_contagem\_sintomas.xlsx: planilha com frequência dos sintomas**

### **3. Exclusão de Doenças Irrelevantes: 03\_excluir\_sintomas.py**

**Essa etapa realizou uma filtragem seletiva das doenças que não seriam utilizadas no modelo de IA, mantendo apenas as classes relevantes.**

#### **Procedimentos:**

**Normalização da coluna: “tipo\_de\_doenca”**

**Conversão de acentuação, letras minúsculas e underscores.**

**Definição de doenças a serem excluídas:**

**endocrina, mamaria, ocular, oral, reprodutiva, sistemica.**

**Filtragem do DataFrame**

**Linhas com essas doenças foram removidas, reduzindo ruído na base.**

#### **Saída final:**

**03\_tabela\_filtrada.xlsx: base refinada contendo apenas doenças pertinentes à classificação.**

## 4. Agrupamento de Classes de Doenças: 04\_agrupamento\_classes.py

Com a base filtrada, esta etapa agrupou doenças semelhantes em grandes categorias, facilitando o aprendizado do modelo classificatório.

### Etapas:

Normalização do texto da coluna

tipo\_de\_doenca.

Criação de um dicionário de agrupamento

Os tipos relacionados foram unificados:

cardiovascular, sanguínea, hematológica -> cardiovascular\_hematologica

neurológica, musculoesquelética -> neuro\_musculoskeletal

renal, reprodutiva, mamária -> urogenital

[...]

Geração da nova coluna: “classe\_doença”

Responsável por armazenar a classe geral de cada registro.

### Saída final:

04\_tabela\_agrupada.xlsx: tabela já normalizada com a coluna de classes adicionada.



## 5. Geração do primeiro arquivo final: 05\_tabela\_final.csv

Após a execução do script “04\_agrupamento\_classes.py”, foi gerado o arquivo “04\_tabela\_agrupada.xlsx”, contendo as informações já organizadas e com a nova coluna “classe\_doenca”, responsável por agrupar doenças semelhantes em uma única categoria.

No entanto, ainda existiam três colunas que se tornaram redundantes após a criação dessa nova classificação: “doenca”, “tipo\_de\_doenca” e “causa”.

Essas colunas foram removidas manualmente da planilha, uma vez que seus dados já estavam representados de forma consolidada pela coluna “classe\_doenca”.

### Resultado:

Com essa limpeza final, obteve-se uma versão otimizada e padronizada da base de dados, que foi então exportada como “05\_tabela\_final.csv”, servindo como entrada principal para as próximas etapas do fluxo de processamento e análise.

# Análises Exploratórias da Base de Dados: analise\_pos05\_tabela\_final.py

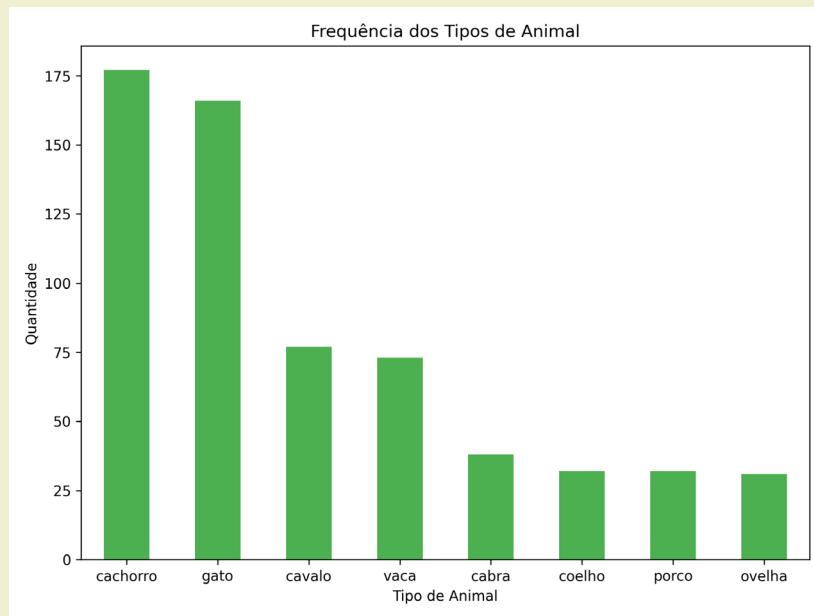
Após a geração do arquivo “05\_tabela\_final.csv”, foi realizada uma etapa de análise exploratória dos dados, feita no script: “analise\_pos05\_tabela\_final.py”.

Essa fase teve como objetivo compreender melhor o comportamento da base que será utilizada no treinamento da Inteligência Artificial da PetDex, identificando padrões, inconsistências e possíveis redundâncias.

Foram conduzidas as seguintes análises principais:

## Frequência dos tipos de animais

Essa visualização permite identificar quais tipos de animais possuem maior representatividade na base, o que é fundamental para avaliar se o modelo de aprendizado de máquina terá amostragem equilibrada entre as classes.



O gráfico resultante evidenciou que há 8 tipos de animais diferentes, sendo alguns com ocorrência significativamente maior que outros, o foco da nossa classificação: cachorros e gatos.

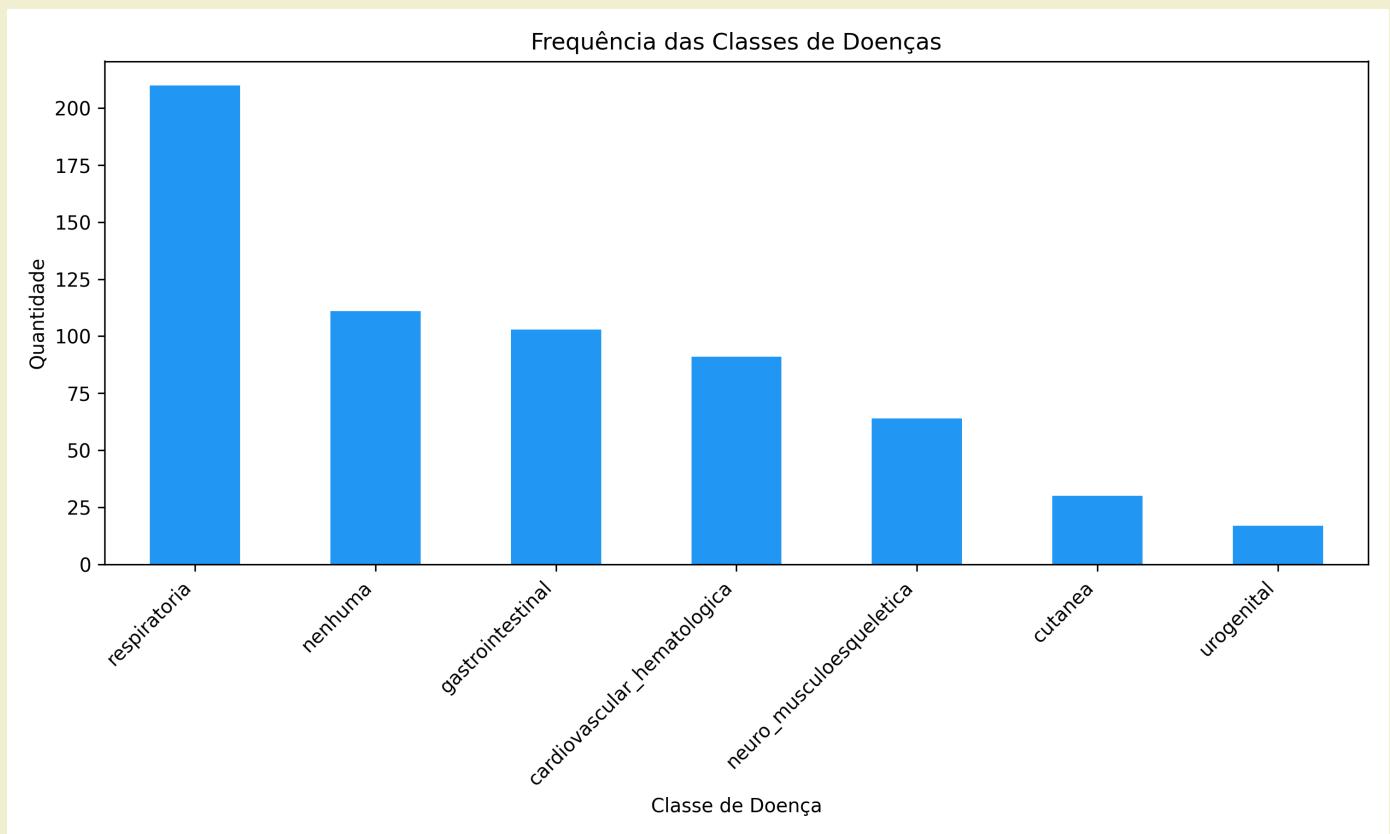


# Análises Exploratórias da Base de Dados: analise\_pos05\_tabela\_final.py

## Frequência das classes de doenças

Com base na coluna `classe_doenca` (criada na etapa anterior de agrupamento), foi gerado um segundo gráfico de barras mostrando a quantidade de ocorrências de cada classe.

Essa análise ajuda a entender quais doenças são mais comuns e se existe algum desequilíbrio entre as classes, o que pode impactar diretamente a performance do modelo de classificação.



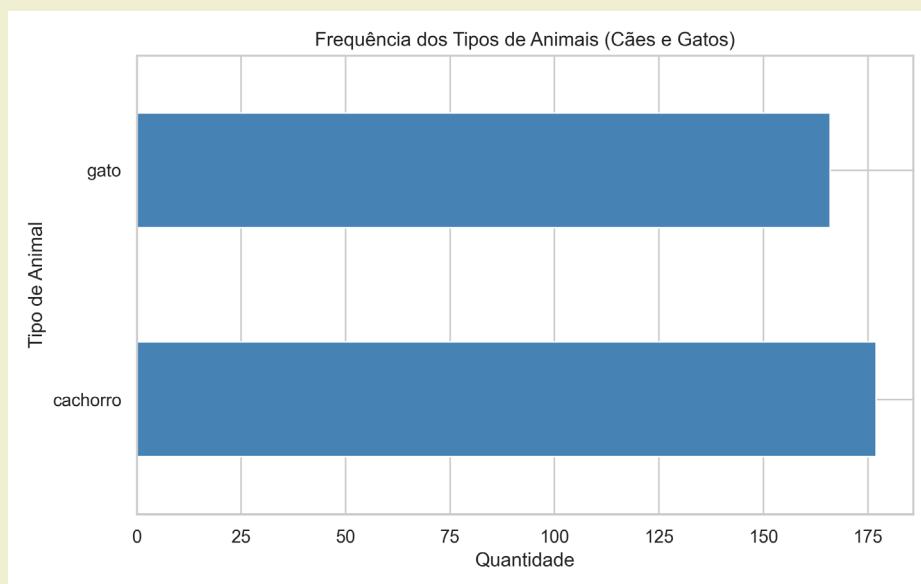
No total, foram identificadas 7 classes de doenças, agrupando condições de natureza semelhante.

# Filtragem: criação da tabela apenas com Cães e Gatos: `filtro_cachorros_e_gatos.py`

O objetivo foi selecionar da base consolidada (`05_tabela_final.csv`) apenas as instâncias referentes aos animais mais relevantes para o projeto, cachorros e gatos, resultando em uma nova tabela dedicada para análises específicas: `06_tabela_cachorros_gatos.csv`.

## Por que esse passo é importante:

- Permite análises mais direcionadas e relevantes, sendo elas a comparação entre cães e gatos.
- Evita ruído e vieses causados por espécies com pouca representação na base.
- Facilita a criação de modelos de classificação específicos ou a comparação de padrões entre as duas espécies.



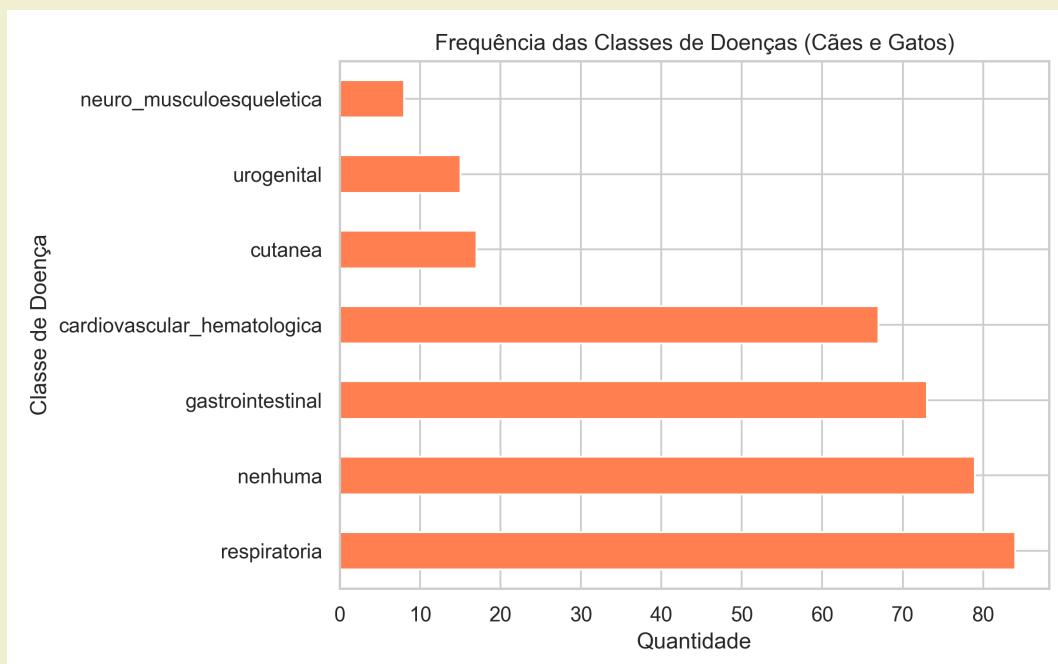
Após o filtro vamos trabalhar agora com os dados de 177 cachorros e 166 gatos. Observa-se que os cães possuem leve predominância em relação aos gatos, mas ambos estão bem representados, garantindo equilíbrio e diversidade suficientes para análises comparativas.



# Análise Visual dos Dados Filtrados: analise\_cachorro\_e\_gatos.py

A partir do novo subconjunto, foram executadas diversas análises de frequência e correlação, representadas em gráficos e matrizes que permitem observar de forma intuitiva a distribuição das variáveis e suas possíveis relações.

## Frequência das Classes de Doenças:



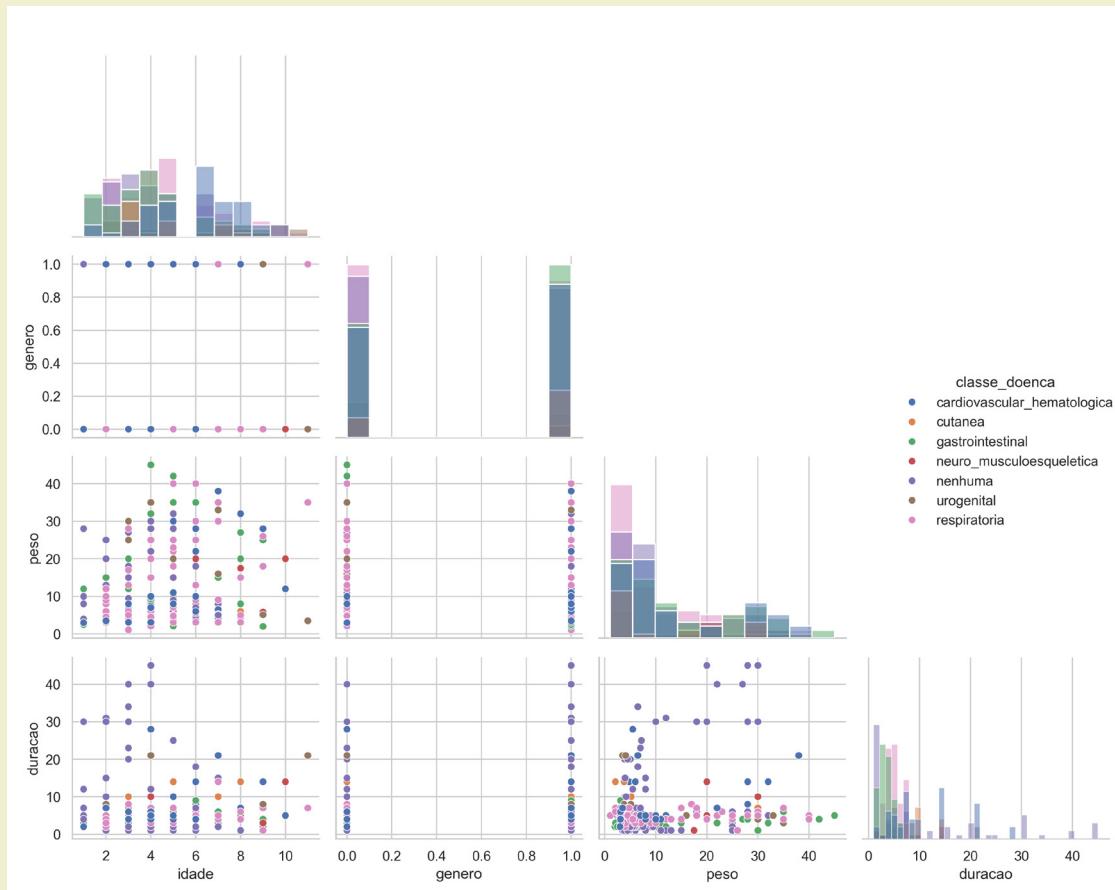
O gráfico mostra a distribuição das classes de doenças entre os cães e gatos do conjunto filtrado, onde é possível observar que as doenças respiratórias, gastrointestinais e cardiovasculares/hematológicas são as mais recorrentes, seguidas por registros classificados como “nenhuma” (ou seja, sem sintomas identificados).

Classes como cutâneas, urogenitais e neuromusculoesqueléticas aparecem em menor frequência, representando uma parcela mais específica dos casos.

Esses resultados indicam que a maior parte das ocorrências envolve condições de saúde comuns e de diagnóstico frequente em animais domésticos, como tosse, dificuldade respiratória e distúrbios digestivos, o que reforça a importância do monitoramento contínuo da saúde desses animais.

# Análise Visual dos Dados Filtrados: analise\_cachorro\_e\_gatos.py

## Matriz de Dispersão (Pairplot)



A matriz de dispersão, também chamada de pairplot, apresenta a relação entre variáveis numéricas como idade, peso, duração e gênero, segmentadas pelas classes de doença.

Cada ponto representa um animal, colorido de acordo com sua respectiva classe de doença. É possível notar algumas tendências visuais, como:

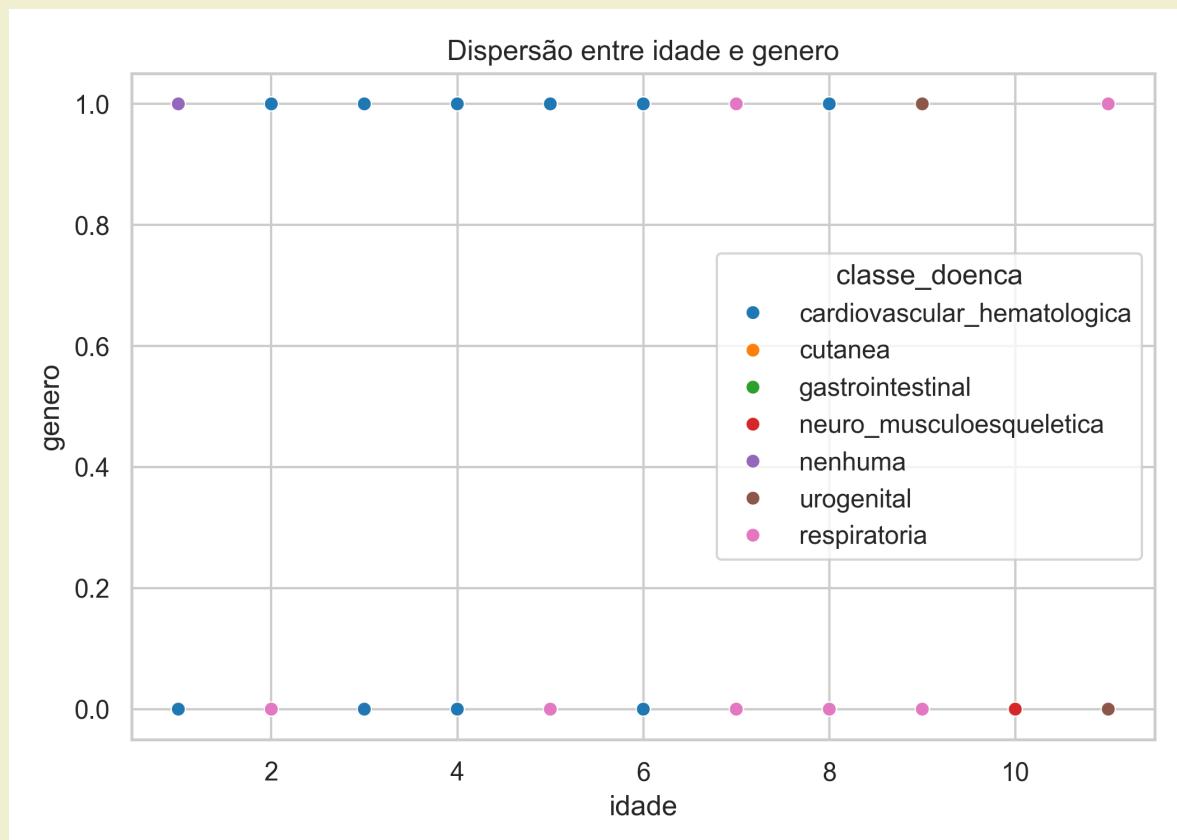
- Certas doenças se concentrando em faixas etárias específicas (por exemplo, doenças respiratórias em animais mais jovens);
- Variação de peso e duração associada a tipos de doenças distintas;
- Distribuição equilibrada entre gêneros dentro das classes analisadas.

Essa visualização oferece uma visão global das relações entre variáveis e permite identificar possíveis agrupamentos ou padrões naturais nos dados.



# Análise Visual dos Dados Filtrados: analise\_cachorro\_e\_gatos.py

## Gráfico de Dispersão (Scatter Plot)



O gráfico de dispersão individual evidencia a relação direta entre idade e gênero, novamente segmentada pelas classes de doença.

Embora o gênero seja uma variável categórica binária (codificada como 0 e 1), o gráfico é útil para verificar a distribuição das doenças entre machos e fêmeas.

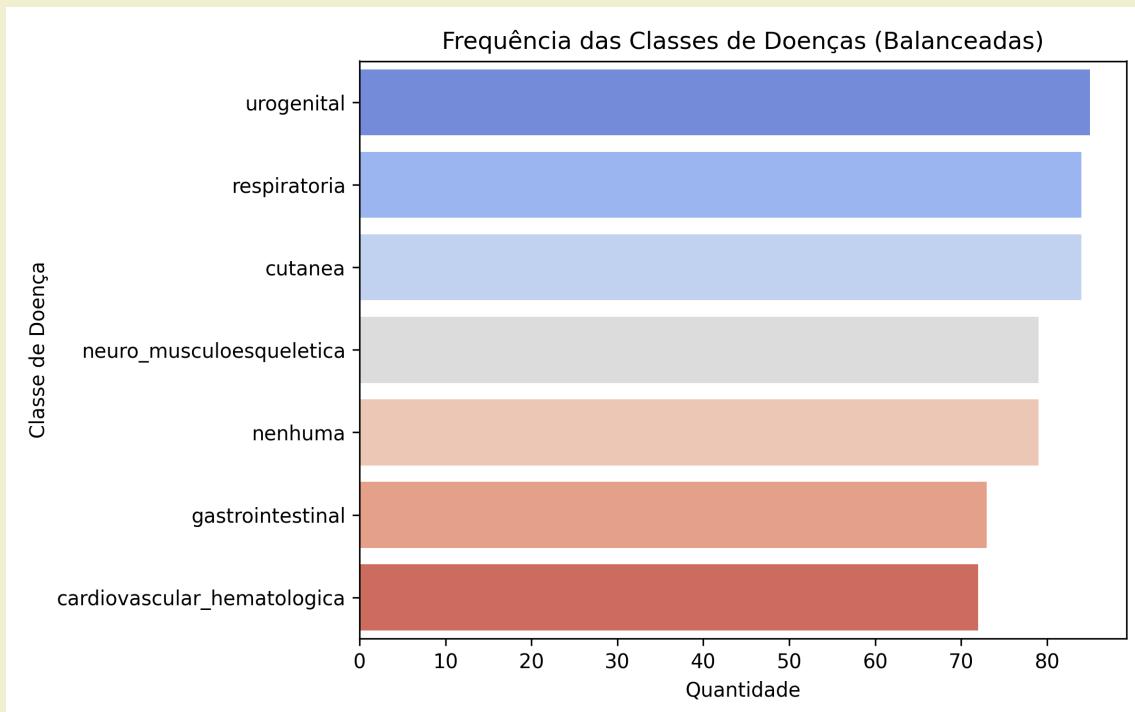
Essa visualização confirma que as doenças estão distribuídas de forma equilibrada entre machos e fêmeas, não havendo indicativos de que o gênero influencie significativamente na prevalência das condições.

# Balanceamento das Classes de Doenças: balanceamento01.py

Durante a análise inicial dos dados, foi possível observar um desequilíbrio significativo entre as classes de doenças. Algumas categorias, como respiratória, nenhuma e gastrointestinal, possuíam muitos registros, enquanto outras, como cutânea, urogenital e neuro\_musculoesquelética, apresentavam baixa representatividade.

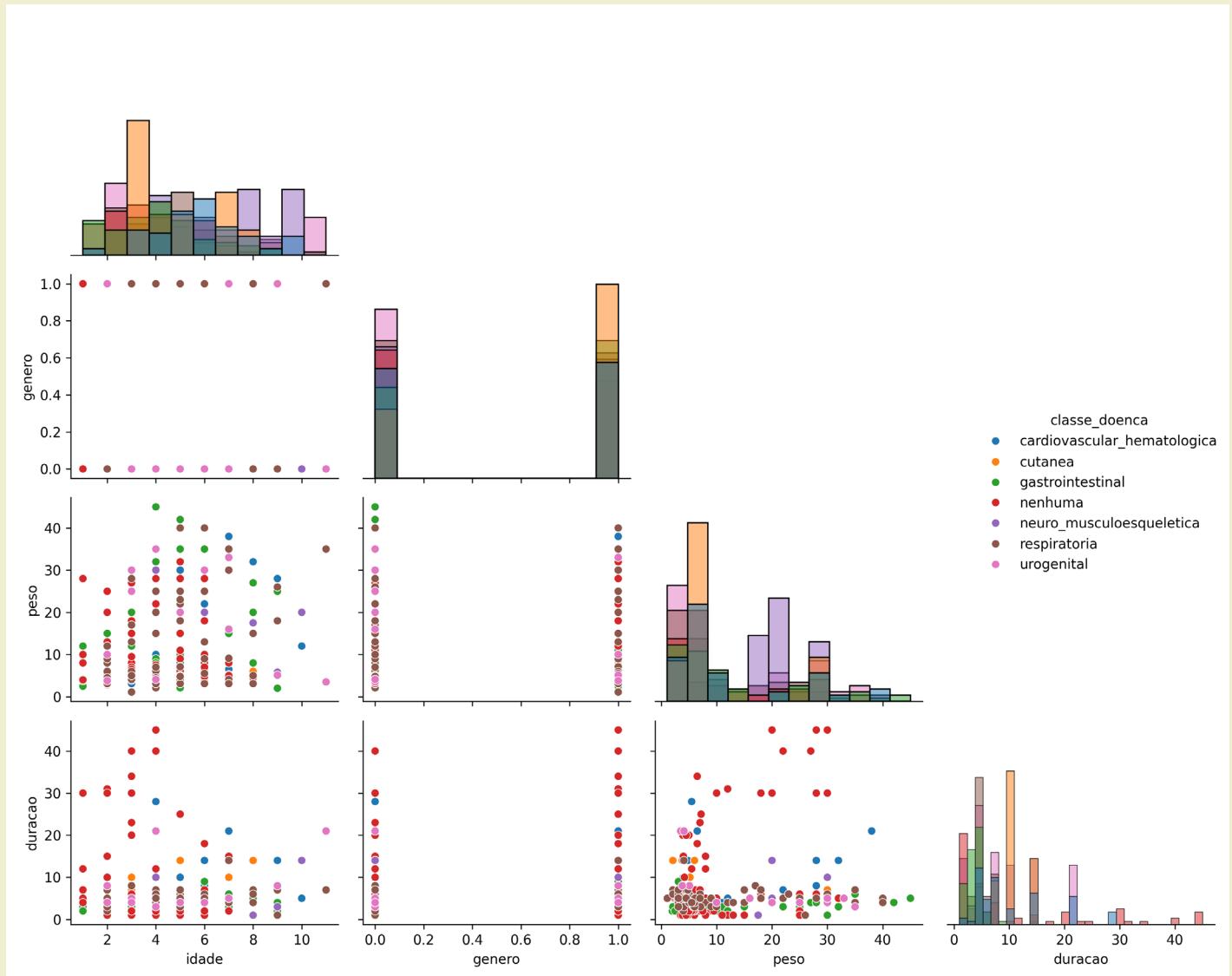
Esse tipo de desbalanceamento pode prejudicar análises estatísticas e o desempenho de modelos de Inteligência Artificial, uma vez que as classes minoritárias tendem a ser negligenciadas pelos algoritmos.

Para corrigir isso, foi aplicado um processo de balanceamento dos dados, que consistiu em aumentar a quantidade de registros das classes minoritárias por meio de oversampling aleatório, replicando aleatoriamente registros existentes dessas classes até que o número de ocorrências ficasse mais próximo das demais, sem alterar as classes que já estavam equilibradas.



# Balanceamento das Classes de Doenças: analise\_posbalanceamento.py

## Matriz de Dispersão (Pairplot)



O objetivo desse procedimento foi garantir uma distribuição mais justa e representativa das doenças, permitindo uma análise mais confiável e um melhor treinamento de modelos de aprendizado de máquina.

# Treinamento dos Modelos de Classificação: treinamento\_01.py

Após a etapa de balanceamento dos dados, iniciou-se o processo de treinamento dos modelos de classificação que serão utilizados pela Inteligência Artificial da PetDex. O objetivo desta fase foi avaliar diferentes algoritmos supervisionados, comparar seus desempenhos e exportar os melhores modelos para o formato PMML, possibilitando sua integração com a API e com o aplicativo mobile.

O treinamento foi realizado utilizando o arquivo:  
**06\_tabela\_cachorros\_gatos\_balanceado\_variante.csv**  
contendo apenas registros de cães e gatos.

## Pré-processamento

O código realizou automaticamente:

- Separação entre atributos (X) e classe alvo (y).
- Identificação de colunas numéricas e categóricas.
- Padronização dos atributos numéricos com StandardScaler.
- Codificação das variáveis categóricas com OneHotEncoder.
- Criação de um PMMLPipeline

A base foi dividida em:

- 80% para treinamento
- 20% para validação

## Modelos avaliados

Foram treinados e comparados seis algoritmos:

1. Regressão Logística (LR)
2. Linear Discriminant Analysis (LDA)
3. K-Nearest Neighbors (KNN)
4. Decision Tree (CART)
5. Naive Bayes (NB)
6. Support Vector Machine (SVM)

A avaliação no conjunto de treino foi feita utilizando Cross-Validation com 10 folds, garantindo maior robustez estatística na comparação.

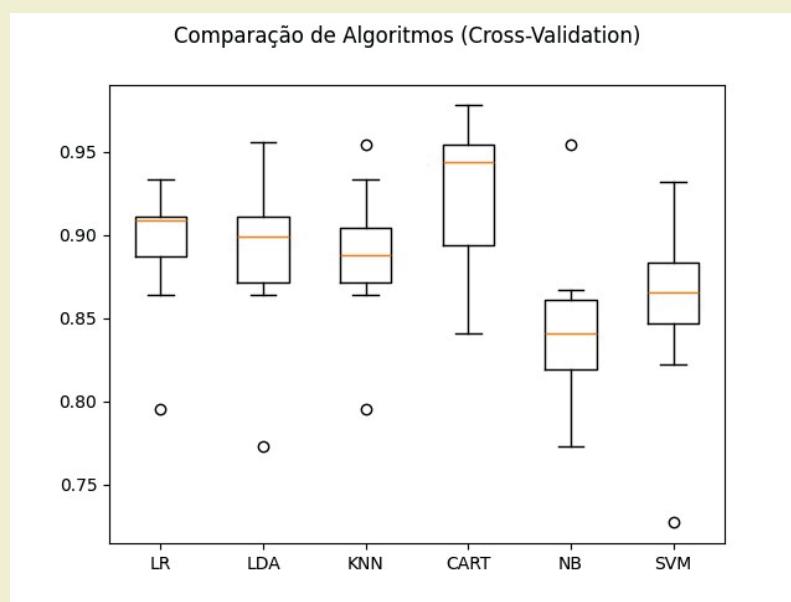
# Resultados do Cross-Validation: treinamento\_01.py

Os resultados obtidos na avaliação de cross-validation foram:

Modelo	Acurácia Média
LR	0.8895
LDA	0.8895
KNN	0.8873
CART	<b>0.9255</b> (melhor desempenho)
NB	0.8423
SVM	0.8581

A análise do conjunto de treino indica que:

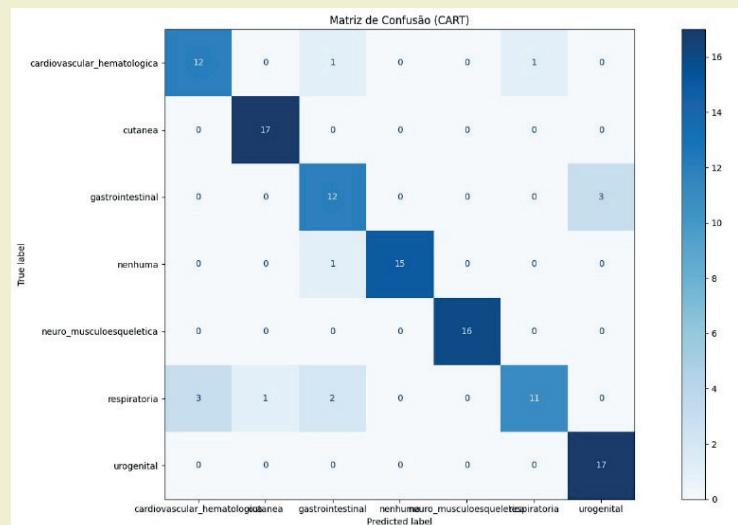
- O modelo Decision Tree (CART) apresentou o melhor desempenho geral, com acurácia superior a 92%.
- LR, LDA e KNN obtiveram resultados muito próximos, acima de 88%.
- O Naive Bayes teve o desempenho mais baixo, porém ainda aceitável.
- O SVM apresentou estabilidade e boa acurácia (85%).



# Avaliação no Conjunto de Validação: treinamento\_01.py

Após a escolha preliminar via cross-validation, cada modelo foi testado no conjunto de 20% reservado para validação, simulando sua performance em novos dados:

Modelo	Acurácia na Validação
LR	0.8571
LDA	0.8571
KNN	0.8571
CART	<b>0.8929</b>
NB	0.8036
SVM	0.8750



O modelo CART confirmou ser o mais eficiente, apresentando novamente o melhor desempenho (89,29%).

## Interpretação dos Resultados

Os relatórios de classificação e as matrizes de confusão mostram que:

- As classes cutânea, neuro\_musculoesquelética, nenhuma, e urogenital apresentam excelente performance na maioria dos modelos.
- As classes respiratória e gastrointestinal foram as que geraram mais erros, possivelmente devido à similaridade de sintomas.
- Mesmo assim todos os modelos, exceto NB, atingiram valores próximos ou superiores a 85% de acurácia.

# Resultados do Cross-Validation: treinamento\_01.py

LR - Acurácia: 0.8571 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.79	0.79	0.79	14
cutanea	0.94	1.00	0.97	17
gastrointestinal	0.90	0.60	0.72	15
nenhuma	1.00	0.94	0.97	16
neuro_musculoesquelética	0.94	1.00	0.97	16
respiratoria	0.79	0.65	0.71	17
urogenital	0.71	1.00	0.83	17
accuracy			0.86	112
macro avg	0.87	0.85	0.85	112
weighted avg	0.87	0.86	0.85	112

KNN - Acurácia: 0.8571 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.67	0.86	0.75	14
cutanea	0.94	1.00	0.97	17
gastrointestinal	0.90	0.60	0.72	15
nenhuma	1.00	0.94	0.97	16
neuro_musculoesquelética	1.00	1.00	1.00	16
respiratoria	0.92	0.65	0.76	17
urogenital	0.70	0.94	0.80	17
accuracy			0.86	112
macro avg	0.87	0.85	0.85	112
weighted avg	0.88	0.86	0.86	112

LDA - Acurácia: 0.8571 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.79	0.79	0.79	14
cutanea	0.94	1.00	0.97	17
gastrointestinal	0.90	0.60	0.72	15
nenhuma	1.00	0.94	0.97	16
neuro_musculoesquelética	0.94	1.00	0.97	16
respiratoria	0.73	0.65	0.69	17
urogenital	0.74	1.00	0.85	17
accuracy			0.86	112
macro avg	0.86	0.85	0.85	112
weighted avg	0.86	0.86	0.85	112

CART - Acurácia: 0.8929 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.80	0.86	0.83	14
cutanea	0.94	1.00	0.97	17
gastrointestinal	0.75	0.80	0.77	15
nenhuma	1.00	0.94	0.97	16
neuro_musculoesquelética	1.00	1.00	1.00	16
respiratoria	0.92	0.65	0.76	17
urogenital	0.85	1.00	0.92	17
accuracy			0.89	112
macro avg	0.89	0.89	0.89	112
weighted avg	0.90	0.89	0.89	112

NB - Acurácia: 0.8036 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.55	0.79	0.65	14
cutanea	1.00	1.00	1.00	17
gastrointestinal	0.73	0.53	0.62	15
nenhuma	1.00	0.75	0.86	16
neuro_musculoesquelética	1.00	1.00	1.00	16
respiratoria	0.75	0.53	0.62	17
urogenital	0.71	1.00	0.83	17
accuracy			0.80	112
macro avg	0.82	0.80	0.80	112
weighted avg	0.83	0.80	0.80	112

SVM - Acurácia: 0.8750 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	1.00	0.71	0.83	14
cutanea	0.94	1.00	0.97	17
gastrointestinal	1.00	0.60	0.75	15
nenhuma	1.00	0.94	0.97	16
neuro_musculoesquelética	1.00	1.00	1.00	16
respiratoria	0.78	0.82	0.80	17
urogenital	0.65	1.00	0.79	17
accuracy			0.88	112
macro avg	0.91	0.87	0.87	112
weighted avg	0.91	0.88	0.87	112

# Exportação dos Modelos em PMML: treinamento\_01.py

Após finalizar o treinamento dos seis algoritmos, cada modelo foi convertido automaticamente para o formato PMML (Predictive Model Markup Language). Esse formato padronizado permite que os modelos sejam utilizados fora do ambiente Python, garantindo portabilidade e integração com diversos sistemas.

## Arquivos PMML Gerados

- `modelo_LR.pmml`
- `modelo_LDA.pmml`
- `modelo_KNN.pmml`
- `modelo_CART.pmml`
- `modelo_NB.pmml`
- `modelo_SVM.pmml`

## Por que usar PMML?

A exportação dos modelos em PMML torna possível:

- Integrar facilmente os algoritmos à API da PetDex;
- Reutilizar modelos em ambientes de produção sem depender do código original;
- Padronizar o fluxo de inferência, permitindo que múltiplos serviços consumam as mesmas previsões;
- Facilitar atualizações futuras, bastando substituir o arquivo PMML para atualizar o modelo.

## Balanceamento 02 e Treinamento dos Modelos: (Todos os Animais)

Para fins de estudo comparativo, foi necessário repetir o processo de preparação e treinamento, agora utilizando a base completa contendo todos os animais, antes do filtro específico para cães e gatos. O objetivo foi verificar como os algoritmos se comportam em um cenário mais amplo e heterogêneo.

### Balanceamento das Classes (`balanceamento02.py`)

Assim como no primeiro experimento, algumas classes de doenças apresentavam quantidades muito diferentes de registros. Para resolver isso, foi aplicado novamente oversampling aleatório, porém com uma variação controlada: cada classe minoritária foi ampliada até ficar dentro de um intervalo de 90% a 110% da maior classe. Isso garante um balanceamento mais natural, preservando a variabilidade dos dados.

O resultado final foi salvo no arquivo:

`06_tabela.todos_animais_balanceado_variante.csv`

### Treinamento 02 – Modelos Aplicados a Todos os Animais (`treinamento_02.py`)

Com a tabela balanceada, os seis modelos estatísticos foram retreinados:

- `modelo_LR(todos_animais).pmml`
- `modelo_LDA(todos_animais).pmml`
- `modelo_KNN(todos_animais).pmml`
- `modelo_CART(todos_animais).pmml`
- `modelo_NB(todos_animais).pmml`
- `modelo_SVM(todos_animais).pmml`

# Resultados do Cross-Validation: treinamento\_02.py

LR - Acurácia: 0.9088 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.97	0.90	0.94	41
cutanea	0.92	1.00	0.96	45
gastrointestinal	0.88	0.82	0.85	45
nenhuma	1.00	1.00	1.00	43
neuro_musculoesquelética	0.85	1.00	0.92	41
respiratoria	0.84	0.64	0.73	42
urogenital	0.89	1.00	0.94	39
accuracy			0.91	296
macro avg	0.91	0.91	0.91	296
weighted avg	0.91	0.91	0.90	296

SVM - Acurácia: 0.7399 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	1.00	0.46	0.63	41
cutanea	0.79	0.82	0.80	45
gastrointestinal	0.85	0.49	0.62	45
nenhuma	0.97	0.91	0.94	43
neuro_musculoesquelética	0.73	0.80	0.77	41
respiratoria	0.49	0.76	0.60	42
urogenital	0.69	0.95	0.80	39
accuracy				0.74
macro avg	0.79	0.74	0.74	296
weighted avg	0.79	0.74	0.74	296

□

CART - Acurácia: 0.9324 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.87	0.98	0.92	41
cutanea	1.00	1.00	1.00	45
gastrointestinal	0.93	0.91	0.92	45
nenhuma	1.00	1.00	1.00	43
neuro_musculoesquelética	0.89	1.00	0.94	41
respiratoria	0.90	0.64	0.75	42
urogenital	0.93	1.00	0.96	39
accuracy			0.93	296
macro avg	0.93	0.93	0.93	296
weighted avg	0.93	0.93	0.93	296

NB - Acurácia: 0.8176 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.66	0.95	0.78	41
cutanea	0.87	1.00	0.93	45
gastrointestinal	0.82	0.71	0.76	45
nenhuma	0.94	0.79	0.86	43
neuro_musculoesquelética	0.80	0.95	0.87	41
respiratoria	0.82	0.33	0.47	42
urogenital	0.89	1.00	0.94	39
accuracy				0.82
macro avg	0.83	0.82	0.80	296
weighted avg	0.83	0.82	0.80	296

KNN - Acurácia: 0.8514 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.86	0.90	0.88	41
cutanea	0.88	1.00	0.94	45
gastrointestinal	0.85	0.73	0.79	45
nenhuma	0.95	0.95	0.95	43
neuro_musculoesquelética	0.81	0.93	0.86	41
respiratoria	0.70	0.45	0.55	42
urogenital	0.85	1.00	0.92	39
accuracy			0.85	296
macro avg	0.84	0.85	0.84	296
weighted avg	0.84	0.85	0.84	296

LDA - Acurácia: 0.9088 Classification Report:				
	precision	recall	f1-score	support
cardiovascular_hematologica	0.89	0.95	0.92	41
cutanea	0.88	1.00	0.94	45
gastrointestinal	0.83	0.87	0.85	45
nenhuma	1.00	1.00	1.00	43
neuro_musculoesquelética	0.89	0.98	0.93	41
respiratoria	0.92	0.57	0.71	42
urogenital	0.97	1.00	0.99	39
accuracy				0.91
macro avg	0.91	0.91	0.90	296
weighted avg	0.91	0.91	0.90	296

Nesta outra situação o modelo CART se manteve como o mais eficiente (93,24%).

# Simulação de Cenário Real e Comparaçāo de Modelos (Amostra de 20 Casos)

Para validar o desempenho dos modelos no contexto exato do aplicativo, foi conduzido um teste comparativo final. O objetivo era responder a uma questão crítica: o modelo treinado com “Todos os Animais” seria mais preciso, ou o modelo especialista treinado apenas com “Gatos e Cachorros” teria um desempenho superior?

## Metodologia

Foi gerado um conjunto de dados de teste focado:

**amostra\_gatos\_cachorros\_20\_linhas.csv**

contendo 20 amostras reais selecionadas aleatoriamente do universo de gatos e cães.

Os 12 arquivos de modelo PMML (os 6 da família "Gato/Cachorro" e os 6 da família "Todos Animais") foram então testados diretamente contra esta amostra. O script de teste (**comparando\_modelos02.py**) foi refinado para conseguir interpretar os padrões de saída de probabilidade e afins distintos encontrados nos PMMLs gerados.

## Ranking Final de Desempenho

Após a execução do teste nos 12 modelos, 9 apresentaram resultados funcionais. O ranking de acurácia foi o seguinte:

Posição	Modelo	Acurácia no Teste	Acertos (de 20)
1º	CART (Gato/Cachorro)	100.00%	20
2º	CART (Todos Animais)	95.00%	19
3º	LR (Gato/Cachorro)	85.00%	17
4º	LR (Todos Animais)	85.00%	17
5º	LDA (Gato/Cachorro)	85.00%	17
6º	LDA (Todos Animais)	85.00%	17
7º	NB (Gato/Cachorro)	80.00%	16
8º	SVM (Gato/Cachorro)	80.00%	16
9º	SVM (Todos Animais)	80.00%	16

# Análise de Compatibilidade e Diagnóstico de Modelos Não-Conclusivos

## Contextualização dos Resultados

Como demonstrado na página 21, nove dos doze modelos treinados completaram o teste de simulação com sucesso, fornecendo métricas de acurácia claras. No entanto, três modelos, ambos da família KNeighborsClassifier (KNN) e o GaussianNB (NB) treinado com todos os animais, não produziram um resultado de acurácia neste teste específico.

É importante ressaltar que os scripts de treino destes modelos (conforme documentado anteriormente) foram executados com sucesso, e os modelos foram validados no ambiente Scikit-learn. As falhas observadas ocorreram estritamente durante a etapa de previsão utilizando os arqui-

## Diagnóstico e Conclusão das Falhas no Teste PMML

A falha dos três modelos (ambos KNN e o NB Todos Animais) no teste não se deve ao treino, mas a incompatibilidades técnicas da cadeia de ferramentas PMML (`sklearn2pmml` e `pypmml`).

- 1. Modelos KNN (ambos):** A exportação para PMML falhou em gerar arquivos compatíveis. O KNN (Gato/Cachorro) foi gerado esperando a própria coluna de resposta como um dado de entrada. O KNN (Todos Animais) foi gerado sem a predição final, retornando apenas os "vizinhos" (ex: `neighbor(1)`), impossibilitando o cálculo da acurácia.
- 2. NB (Todos Animais):** O modelo gerou um valor inválido (`NaN`) durante a previsão na amostra. Isso causou um erro de cálculo ('< not supported'), indicando que o motor PMML não conseguiu processar um "edge case" estatístico deste modelo específico.

Em resumo, a falha ocorreu na "tradução" e "execução" dos modelos PMML, não na sua lógica de treino.

# Conclusão da Análise e Modelo Selecionado

## O Veredito da Análise Comparativa

Após a execução da bateria de testes de validação cruzada e, de forma mais decisiva, da simulação de cenário real, chegamos a uma conclusão clara e definitiva.

**O modelo CART (Gato/Cachorro), a Árvore de Decisão treinada exclusivamente no dataset de cães e gatos, foi selecionado como a solução de Inteligência Artificial para o projeto PetDex.**

## Justificativa Estratégica da Escolha

A seleção deste modelo não se baseia apenas em sua pontuação, mas em um conjunto de fatores estratégicos que o validam como a melhor solução para o problema de negócio:

- **Desempenho Superior:** Foi o único algoritmo a atingir 100% de acurácia no teste de simulação com 20 amostras, provando sua eficácia no cenário exato de uso do aplicativo.
- **Validação da Hipótese "Especialista":** O teste comprovou que o modelo "especialista" (focado no público-alvo) foi significativamente mais preciso que seu equivalente "generalista" (95%), validando a estratégia de segmentação de dados.
- **Interpretabilidade e Confiança:** Sendo uma Árvore de Decisão, o modelo CART oferece alta interpretabilidade. Seus caminhos lógicos podem ser auditados e validados, um fator crucial para um sistema de apoio ao pré-diagnóstico.

## Da Análise à Aplicação: O Modelo em Ação

O objetivo deste relatório foi analisar e validar o "cérebro" do sistema PetDex. Com a seleção do arquivo `modelo_CART.pmml` como o componente central de IA, a análise de machine learning está concluída.

Este modelo PMML foi então integrado ao backend do aplicativo. As páginas a seguir demonstram o funcionamento da solução completa, ilustrando a jornada do usuário desde a inserção de sintomas no aplicativo móvel até a resposta final fornecida pela Inteligência Artificial.

# Aplicação em Funcionamento

**Checkup Inteligente**  
Descubra o que o seu pet pode estar sentindo

Responda algumas perguntas rápidas sobre os sintomas observados e deixe a inteligência da PetDex analisar os dados para identificar possíveis problemas de saúde.

A nossa análise poderá indicar se há sinais relacionados a:

- Sistema cardiovascular e hematológico
- Problemas de pele (cutâneas)
- Distúrbios gastrointestinais
- Problemas neurológicos ou musculoesqueléticos
- Alterações respiratórias
- Condições do trato urinário ou genital

Ou indicar que está tudo bem! 🐾

Iniciar

Essa análise tem caráter informativo e não substitui a avaliação de um médico veterinário.

**Comportamento e Rotina**

Uno está agitado ou mais inquieto que o normal?

Sim  Não

Você notou letargia?  
Desânimo, cansaço ou dorme mais que o normal?

Sim  Não

Uno demonstra fraqueza ou dificuldade em se levantar?

Sim  Não

Está andando em círculos, sem motivo aparente?

Sim  Não

Está rangendo os dentes com frequência?

Sim  Não

Apresenta lambbedura excessiva em alguma parte do corpo?

Sim  Não

**Continuar**

← Voltar

**Alimentação e Digestão**

Uno perdeu o apetite recentemente?

Sim  Não

Houve aumento no apetite, comendo mais que o normal?

Sim  Não

Está vomitando com frequência?

Sim  Não

Apresenta diarreia ou fezes muito moles?

Sim  Não

Você notou perda de peso sem motivo aparente?

Sim  Não

Uno está com sinais de desidratação?  
Boca, olhos ou nariz secos, urina escura e em menor quantidade, boca quente.

Sim  Não

**Continuar**

← Voltar

**Informações do Pet**

Uno ♂ 52 BPM  
Conectado • 96%

Inicio Saúde Checkup Localização

# Aplicação em Funcionamento

The application displays three parallel symptom checklists:

- Respiração e Circulação (Respiration and Circulation):**
  - Uno está com tosse? (Sim  Não )
  - Tem dificuldade para respirar ou respiração ofegante em repouso? (Sim  Não )
  - Você notou roncos ou barulhos diferentes ao respirar? (Sim  Não )
  - Está espirrando com frequência? (Sim  Não )
  - A língua ou gengivas estão azuladas? (Sim  Não )
  - Uno parece ter febre?  
Aparenta estar com o corpo mais quente, especialmente as orelhas. (Sim  Não )
- Movimento e Coordenação (Movement and Coordination):**
  - Uno tem dificuldade para se locomover? (Sim  Não )  
Manca ou evita andar?
  - Demonstra dor ao ser tocado ou ao se mover? (Sim  Não )
  - Você percebeu espasmos musculares?  
Tremores involuntários (Sim  Não )
  - Uno já teve algum desmaio recentemente? (Sim  Não )
  - Há inchaços visíveis em alguma parte do corpo? (Sim  Não )
- Pele, Orelhas e Pelos (Skin, Ears and Fur):**
  - Há problemas na pele, como feridas, irritações ou manchas? (Sim  Não )
  - Uno está com coceira constante? (Sim  Não )
  - Há perda de pelos excessiva ou em áreas específicas? (Sim  Não )
  - Há cera excessiva nas orelhas ou mau cheiro? (Sim  Não )
  - Você notou suor alterado?  
Áreas úmidas ou odor incomum (Sim  Não )
  - Está com salivação maior que o normal? (Sim  Não )

Each screen includes a "Continuar" (Continue) button at the bottom right and a "Voltar" (Back) button at the bottom left.

Below the checklists are three cards showing pet health data:

- Uno ♂ 52 BPM**  
Conectado   
96%  
Bateria
- Uno ♂ 52 BPM**  
Conectado   
96%  
Bateria
- Uno ♂ 52 BPM**  
Conectado   
96%  
Bateria

Each card includes icons for Início (Home), Saúde (Health), Checkup (Checkup), and Localização (Location).

# Aplicação em Funcionamento

23:06 73

## Sintomas Específicos

Há secreção nasal?  
Corrimento pelo nariz

Sim  Não

Há secreção ocular?  
Olhos lacrimejando ou com crostas

Sim  Não

Uno demonstra dificuldade para urinar?

Sim  Não

Qual é a duração dos sintomas?

4 dias

Enviar Respostas

← Voltar

Uno ♂ 52 BPM  
Conectado • 96%

Inicio Saúde Checkup Localização

23:06 73

## Resultado da Análise

Segundo suas respostas ao questionário, a nossa análise identificou

### Problemas neuro-musculoesqueléticos

Nossa análise não substitui uma visita ao veterinário.

Uno ♂ 52 BPM  
Conectado • 96%

Início Saúde Checkup Localização

# Conclusão Geral

## A Solução PetDex

As páginas anteriores demonstraram a jornada completa do projeto PetDex: da análise de dados brutos à seleção detalhada de um modelo de IA até a sua implementação em um aplicativo funcional.

As capturas de tela do aplicativo móvel não são apenas ilustrações, elas representam o ponto de encontro onde os três pilares deste semestre convergem:

1. A **IA Classificadora (CART)**, atuando como o cérebro preditivo.
2. A **Arquitetura de Nuvem (Azure, CI/CD, MongoDB Atlas)**, servindo como o sistema nervoso que conecta e automatiza o fluxo de dados em tempo real.
3. O **Aplicativo (Flutter)**, funcionando como a interface acessível e intuitiva que entrega valor diretamente ao tutor.

## Objetivos Atingidos

Este relatório validou com sucesso a hipótese central do projeto: é viável criar um ecossistema de baixo custo e alta eficiência que utiliza Inteligência Artificial para promover o cuidado preventivo de animais de estimação.

Concluímos a análise provando que um modelo "especialista" (treinado apenas em cães e gatos) não só era viável, como apresentou um desempenho superior (100% de acurácia no teste simulado), justificando plenamente sua seleção.

## Encerramento

O projeto PetDex, portanto, encerra este ciclo de análise como uma solução completa e validada. O sistema, desde a coleta de sintomas no frontend em Flutter até a inferência do modelo PMML no backend em Python, demonstrou ser robusto, escalável e pronto para cumprir sua missão: transformar dados em tranquilidade e cuidado contínuo.

