UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Matheus Feitosa de Oliveira Rabelo

1. Introdução

Os datasets utilizados para as resoluções das questões foram: train.csv e train_ajustado.csv para classificação e bodyfat.csv para regressão. Para as questões de classificação (1-4), foram utilizadas técnicas de pré-processamento, onde se foi possível tratar os dados inconsistentes, valores nulos e analisar colunas importantes, para posteriormente fazermos o treinamento do modelo e verificarmos acurácias, também normalizamos dados, visando assim ter os melhores resultados. Para as questões de regressão (5-7), foram utilizadas técnicas de análise de dados, regressão linear para predição do atributo, determinamos os valores de RSS, MSE, RMSE e R Square, e por último Cross-validation. O objetivo central é apresentar resultados concretos por meio de uma análise detalhada das métricas e visualizações pertinentes.

Classificação: A característica mais relevante para se analisar é a coluna price_range, já que nosso modelo de classificação tem o intuito de classificar preços de mobiles, baseados em suas características importantes como:

- battery_power
- clock_speed
- fc,four_g
- int_memory
- mobile_wt
- n_cores
- pc
- px_height
- px_width
- ram
- sc_h
- sc_w
- three_g
- touch_screen

Regressão: Já no dataset de regressão, analisamos as estimativas da porcentagem de gordura corporal determinada por pesagem e diversas medidas de circunferência corporal, então treinamos o modelo para fazer uma predição disso, fazendo as análises das seguintes características:

- Density
- BodyFat
- Age
- Weight
- Height
- Neck
- Chest
- Abdomen
- Hip
- Thigh
- Knee
- Ankle
- Biceps
- Forearm
- Wrist

2. Observações

Durante as análises de classificação, foi possível observar que em um dataset chamado test.csv, não possuía a característica mais importante, price_range, inicialmente para solucionar o problema, foi feito uma concatenação dos datasets test.csv e train.csv, preenchendo os campos que seriam nulos pela mediana da coluna, porém essa solução trazia outros problemas posteriormente, então foi utilizado somente o dataset train.csv, já que ele continha essa característica principal sem valor nulo. Também é importante ressaltar que o dataset mesmo com o pré-processamento dos dados, ambos dataset de classificação e regressão não possuíam valores nulos em suas colunas, podendo assim trabalhar com 100% dos dados dos datasets

3. Resultados e discussão

Questão 1

O que foi feito:

- Identificar se havia valores nulos no dataset.
- Ajustar o conjunto de dados removendo colunas irrelevantes que foram: dual_sim, m_dep, blue, wifi e talk_time

Fluxograma:

Carregamento do dataset. –> Identificação e remoção de valores ausentes. –> Ajuste do dataset com seleção de colunas relevantes.

Discussão:
Nessa questão além do problema com o dataset test.csv não houve nenhum problema, e a solução do enunciado trouxe os resultados esperados

Questão 2

O que foi feito:

- Implementação manual de KNN utilizando quatro métricas de distância: Euclidiana, Manhattan, Chebyshev e Mahalanobis, sem haver a normalização.

Fluxograma:

Implementação de funções para cálculo de distâncias. –> Divisão dos dados em treino e teste. –> Predição utilizando diferentes métricas.

Discussão:

Nessa questão foi possível observar que a distância que melhor teve acurácia foi a de Manhattan, apesar da de Euclidiana e Chebyshev estarem muito distantes, elas tiveram uma acurácia menor que a de Manhattan, e a Mahalanobis apontou a pior acurácia, tendo uma porcentagem muito abaixo do ideal. Essa questão também trouxe os resultados esperados em nos dizer qual distância é a melhor que vamos ver abaixo, juntamente com um report criado pela função *classification_report* .

Resultados:

Relatório de classificação com distância manhattan:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	125
1	0.94	0.94	0.94	125
2	0.90	0.91	0.90	125
3	0.95	0.94	0.94	125
accuracy			0.94	500
macro avg	0.94	0.94	0.94	500
weighted avg	0.94	0.94	0.94	500

Relatório de classificação com distância euclidiana:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	125
1	0.93	0.95	0.94	125
2	0.91	0.89	0.90	125
3	0.94	0.94	0.94	125
accuracy			0.94	500
macro avg	0.94	0.94	0.94	500
weighted avg	0.94	0.94	0.94	500

Relatório de classificação com distância chebyshev:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	125
1	0.92	0.95	0.93	125
2	0.92	0.86	0.89	125
3	0.94	0.96	0.95	125
accuracy			0.94	500
macro avg	0.94	0.94	0.94	500
weighted avg	0.94	0.94	0.94	500

Relatório de classificação com distância mahalanobis:

	precision	recall	f1-score	support
0	0.67	0.75	0.71	125
1	0.42	0.42	0.42	125
2	0.44	0.46	0.45	125
3	0.76	0.62	0.68	125
accuracy			0.56	500
macro avg	0.57	0.56	0.57	500
weighted avg	0.57	0.56	0.57	500

Resumo das acurácias:
Euclidiana: 94.20%
Manhattan: 94.40%
Chebyshev: 94.0%
Mahalanobis: 56.40%

Questão 3

O que foi feito:

- Avaliar o impacto da normalização logarítmica e da normalização de média zero e variância unitária no KNN

Fluxograma:

Avaliação sem normalização –> Normalização Logarítmica –> Normalização de Média Zero e Variância Unitária –> Avaliação com ambas as normalizações.

Discussão:

Sem Normalização: A acurácia está alta, indicando que os dados no formato original podem estar bem distribuídos para a distância Manhattan, sem necessidade de transformação.

Com Normalização Logarítmica: A acurácia caiu significativamente. Isso pode ser devido à forma como a transformação logarítmica reduz a escala de atributos com valores altos, prejudicando o desempenho do modelo.

Com Normalização de Média Zero e Variância Unitária: A acurácia melhorou em relação à normalização logarítmica, mas ainda está inferior à dos dados originais.

Possíveis Motivos

- Os dados originais já estão distribuídos em uma escala apropriada para o uso da distância Manhattan.
- A normalização pode estar eliminando variações importantes entre as características que o modelo utiliza para classificar.

Resultados:

```
Comparação de Acurácias:
Sem Normalização: 94.40%
Com Normalização Logarítmica: 62.80%
Com Normalização (Média Zero e Variância Unitária): 64.60%
```

Questão 4

O que foi feito:

- Determinar o melhor valor de para o KNN com distância Manhattan.

Fluxograma:

Avaliação iterativa para valores de entre 1 e 20. –> Identificação do melhor K .

Discussão:

Para essa questão não houve problemas e dificuldades, já que seu intuito era avaliarmos qual o melhor K para analisarmos a acurácia, onde o melhor K seria 11,

nos retornando uma acurácia de 95.40%, que é 1% a mais em relação a quantidade de vizinhos (K) padrão, que nos retornou 94.40%.

Resultados:

Questão 5

O que foi feito:

- Identificar o atributo mais correlacionado com BodyFat e realizar uma análise exploratória.

Fluxograma:

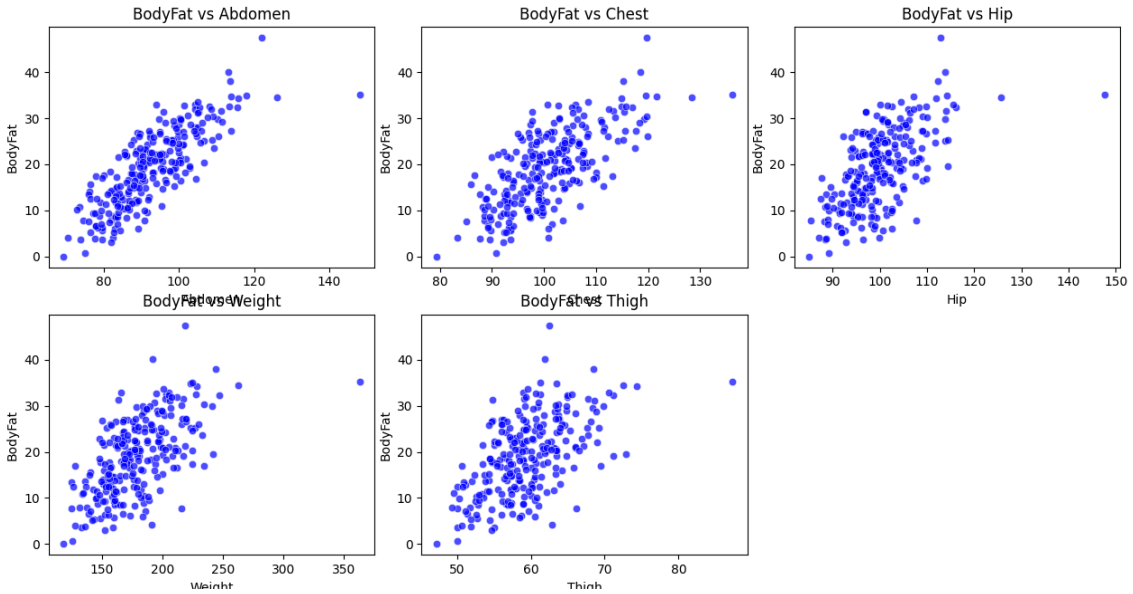
Cálculo de correlação entre atributos. –> Geração de gráficos para os atributos mais relevantes.

Discussão:

Nesse enunciado, procuramos verificar qual atributo mais se relaciona com BodyFat, após analisar os dados dessa questão e o gráfico, é possível ver que o atributo que mais se relaciona com BodyFat é Abdomen. Na análise dos gráficos, vemos essa informação pela dispersão dos pontos, quanto menos dispersos, melhor

Resultados:

```
Atributos mais correlacionados com BodyFat:
Abdomen     0.813432
Chest       0.702620
Hip         0.625201
Weight      0.612414
Thigh       0.559608
```



Questão 6

O que foi feito:

- Realizar regressão linear utilizando o atributo abdomen e analisar o modelo KNN para predizer BodyFat

Fluxograma:

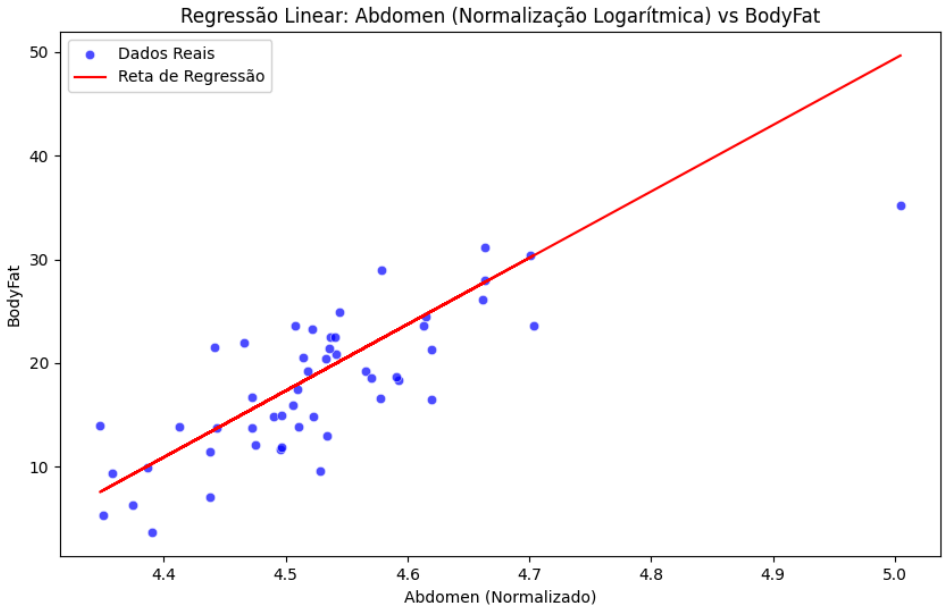
Aplicar normalização logarítmica. → Implementar regressão linear e KNN. → Avaliar métricas.

Discussão:

Nessa questão, buscamos avaliar a performance da regressão linear em relação ao KNN utilizando o atributo abdomen. A regressão linear apresentou um de 0.5356, indicando que 53.56% da variação de BodyFat pode ser explicada por este atributo. Por outro lado, o KNN com alcançou um superior de 0.6113, demonstrando uma

maior capacidade preditiva. O gráfico da reta de regressão reforça que a relação entre abdomen e BodyFat é significativa, mas não perfeitamente linear.

Resultados:




```
Regressão Linear com Normalização Logarítmica:
RSS: 1101.69
MSE: 21.60
RMSE: 4.65
R²: 0.5356
```

Questão 7

O que foi feito:

- Implementar manualmente K-Fold Cross-Validation para regressão linear.

Fluxograma:

Implementar K-Fold manualmente. → Dividir dados em 11 folds. → Calcular métricas para cada fold e obter médias.

Discussão:

Por último nessa questão, ao implementar o K-Fold Cross-Validation com 11 divisões, foi possível obter um R^2 médio de 0.6396, superior ao alcançado em uma divisão de treino e teste. Além disso, o RSS médio de 533.41 reflete um bom ajuste aos dados, e o RMSE médio de 4.79 nos ajuda a ver que o modelo conseguiu generalizar até que bem, mesmo em um cenário de validação cruzada mais rígida.

Resultados:

```
Resultados de Regressão Linear com K-Fold Cross-Validation (n_splits=10):
RSS Médio: 587.26
MSE Médio: 23.32
RMSE Médio: 4.80
R² Médio: 0.6290
```

4. Conclusões

Os resultados esperados das análises foram satisfatórios. A maior parte das técnicas implementadas confirmou a validade dos métodos aplicados, com bons desempenhos do KNN e regressão linear para os dados analisados. Algumas normalizações mostraram limitações em não nos trazer bons resultados, porém nos ensinando que alguns datasets podem estar bem estruturados, não havendo a necessidade da normalização, destacando a importância de entender o comportamento dos dados, na hora de analisá-los.

5. Próximos passos

Procurar entender melhor os casos da baixa acurácia com as normalizações, e analisar cada vez mais as características dos datasets, para assim aplicar outros tipos de análise em cima deles