UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Suziane Brandão Andrade

1. Introdução

- **Classificação: Hotel Reservations Dataset**

  Este Dataset contém informações sobre reservas em hotéis, com um foco específico nos cancelamentos frequentes e nos casos de não comparação dos clientes. Esses cancelamentos e faltas representam um grande desafio para as redes hoteleiras, pois impactam diretamente a capacidade de alocação e a otimização das operações.

  Quando uma reserva é cancelada ou quando um hóspede não aparece, o hotel perde a oportunidade de ocupar o quarto com outro cliente, o que resulta em uma perda de receita.

- **Regressão: Preços de Relógios Inteligentes**

  Este conjunto de dados fornece uma visão detalhada de diversos modelos de smartwatches, permitindo uma análise comparativa entre marcas, características como tamanho da tela, duração da bateria, sistemas operacionais e outras funcionalidades importantes. Através dessa análise, é possível identificar tendências do mercado, comparar o desempenho de diferentes dispositivos e explorar os fatores que influenciam a escolha do consumidor, como preços e funcionalidades.

2. Observações

Algum problema que aconteceu? Alguma observação? Algum imprevisto? Se não houve problemas, deixe em branco.

3. Resultados e discussão

1) Realizar manipulação em um dataset com a biblioteca pandas e realizar o pré-processamento.

Nesta questão foi utilizado o dataset de **Classificação: Hotel Reservations.** Inicialmente foi verificado as informações gerais do dataset, como a dimensão e s

tipos de dados com qual seria trabalhado, usando os argumentos: **shape, dtypes e Info.**

Após essa análise inicial, foi concluído que não havia células vazias ou Nan. Também foi verificado quais colunas eram desnecessárias e retiradas: "Booking_ID", "type_of_meal_plan", "required_car_parking_space", "arrival_year", "arrival_date", 'market_segment_type'

Foi feita uma conversão de duas colunas Usando o Pandas para One-Hot Encoding, a fim de normalizar os dados. Com as modificações foi gerado um novo data frame que será usado posteriormente.

2) Realizar uma classificação utilizando KNN implementado de forma manual

Para implementar o KNN manualmente, segui os seguintes passos:

Com o novo dataset construído na questão anterior, separei o X (feature) e y (Classe),

O objetivo principal, era classificar quais perfis em potencial poderiam cancelar ou mesmo faltar nas reservas de hotéis. Nesse caso, a coluna usada foi "booking_status_Not_Canceled ". Foi feita a divisão manual entre treino e teste, usando a função random para fazer o embaralhamento.

Também foi implementado as seguintes funções:

Manhattan - métrica que calcula a soma das diferenças absolutas entre as coordenadas de dois pontos.

Euclidiana - métrica mais comum e é baseada no Teorema de Pitágoras. Ela calcula a "distância direta" ou "reta" entre dois pontos, como se fosse a distância de uma linha reta.

Chebyshev - é baseada no conceito de movimento em uma série, mas aqui você pode se mover em qualquer direção (horizontal, vertical ou diagonal) e a distância é determinada pelo maior movimento necessário em uma direção.

Mahalanobis - medida estatística que leva em conta a observação entre as variáveis, se baseia também na covariância dos dados.

Ao implementar o KNN, com os dados de treino e teste, os dados foram ordenados por distância, pegando os vizinhos mais próximos, foi feita a contagem das classes vizinhas. Como minha base de dados era bem extensa mesmo após deletar inúmeros dados irrelevantes. Inicialmente fiz um loop para acompanhar a apuração, mas ainda estava demorando. Então como resolução, peguei uma amostragem aleatória de 10%, para usar na classificação com o KNN.

SAÍDA:

```
Tamanho do conjunto de treino: 2539
Tamanho do conjunto de teste: 1089
Acurácia final: 74.93%
Precisão usando Euclidiana: 74.93%
Acurácia final: 76.03%
Precisão usando Manhattan: 76.03%
Acurácia final: 75.67%
Precisão usando Chebyshev: 75.67%
Acurácia final: 78.33%
Precisão usando Mahalanobis: 78.33%

Resultados Finais:
Euclidiana: 74.93%
Manhattan: 76.03%
Chebyshev: 75.67%
Mahalanobis: 78.33%
```

3) Verificar se a normalização interfere nos resultados de sua classificação

Foi feita duas normalizações a Logarítmica e Médio Zero Variância Unitária

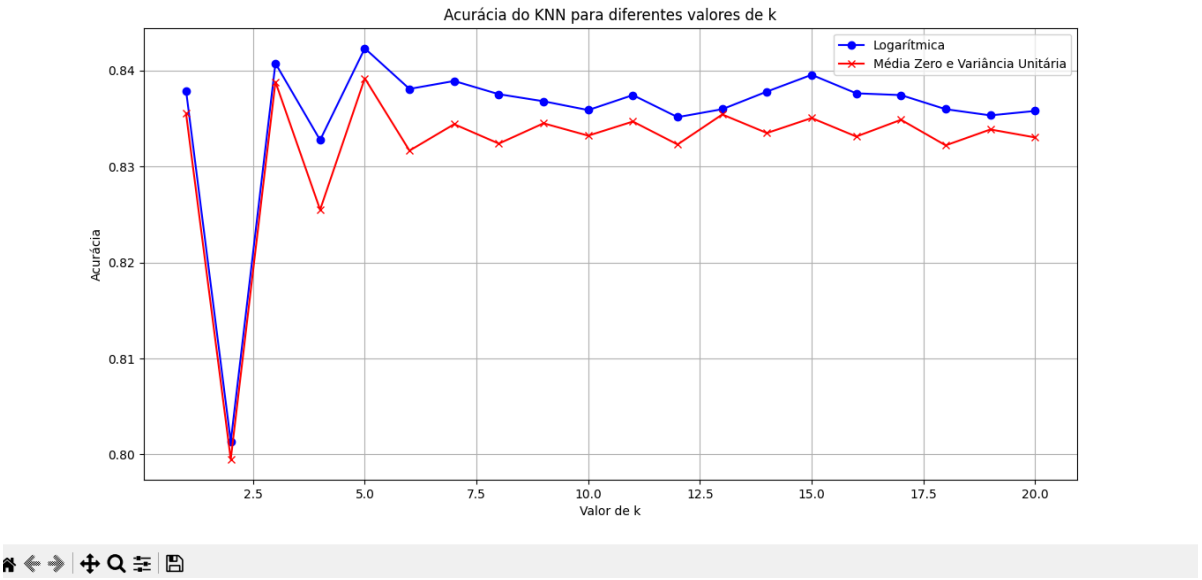
```
[36275 rows x 18 columns]
Acurácia com normalização logaritmica: 83.89%
Acurácia com normalização de média zero e variância unitária: 83.44%

Process finished with exit code 0
```

Com a normalização dos dados a acurácia aumentou significativamente , uma vez que variava entre 73% a 75%, dando um salto para mais de 83%.

4) Buscar saber a melhor parametrização do knn implementado na questão anterior.

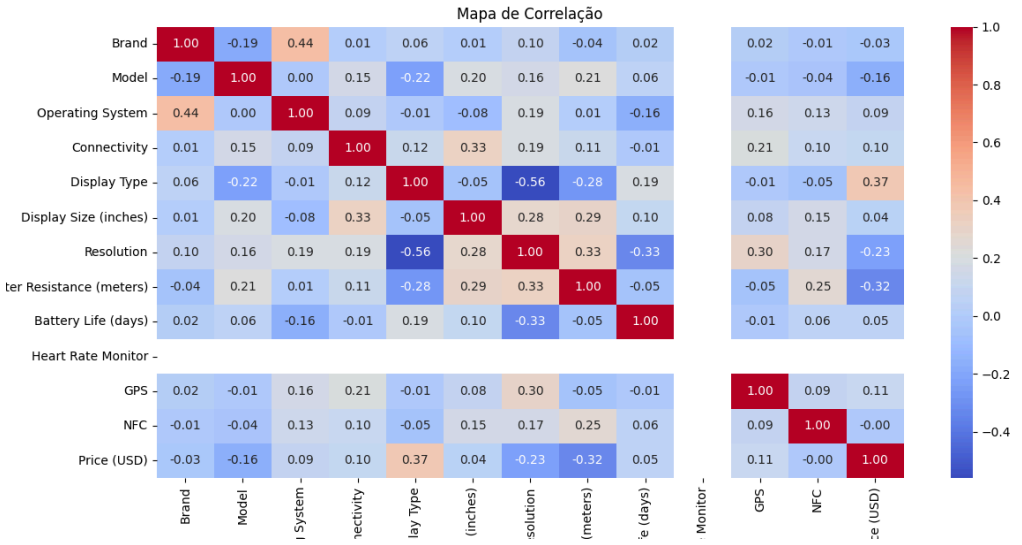
Gráfico plotado onde mostra essa comparação da Acurácia do KNN, comparando as duas normalizações. É possível perceber que por mais próximas, a Logarítmica, quando K = 5, está com a acurácia bem alta.



5) **Observe o dataset de regressão e realize o pré-processamento. Verifique qual atributo será o alvo para regressão no seu dataset e faça uma análise de qual atributo é mais relevante para realizar a regressão do alvo escolhido. Lembre de comprovar via gráfico. Caso necessário remova colunas que são insignificantes, valore NaN também devem ser removidos.**

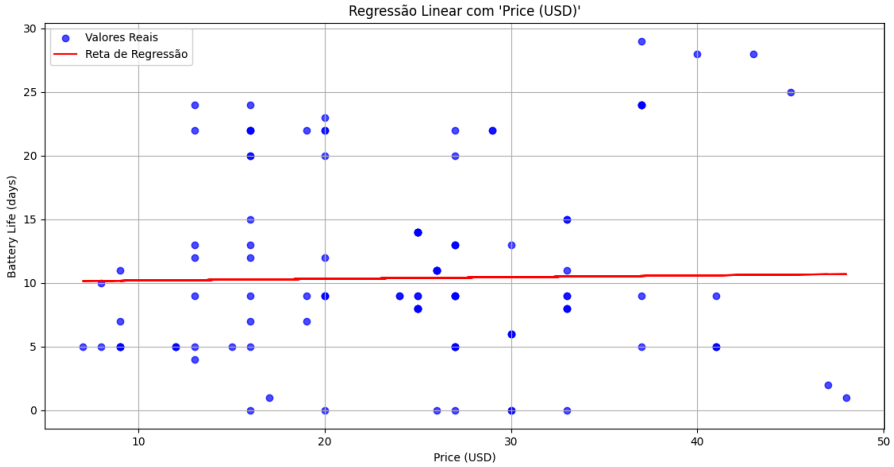
No dataset de **Regressão: Preços de Relógios Inteligentes,** também foi feita uma análise dos dados disponíveis. Ficou constatado alguns valores Nan, que logo foram removidos com comando `dropna`. Também foi necessário fazer a normalização de alguns dados do tipo `object`.

Inicialmente eu gostaria de fazer uma regressão visando saber quanto duraria a bateria com base em atributos do aparelho como modelo e fabricante. Mas ao plotar o mapa de correlação, percebi que a duração de bateria tinha pouquíssima relação com os outros dados dos Smarth Watch. Então mudei para uma regressão linear usando **Display Size (inches)** ou **Brand**(marca).

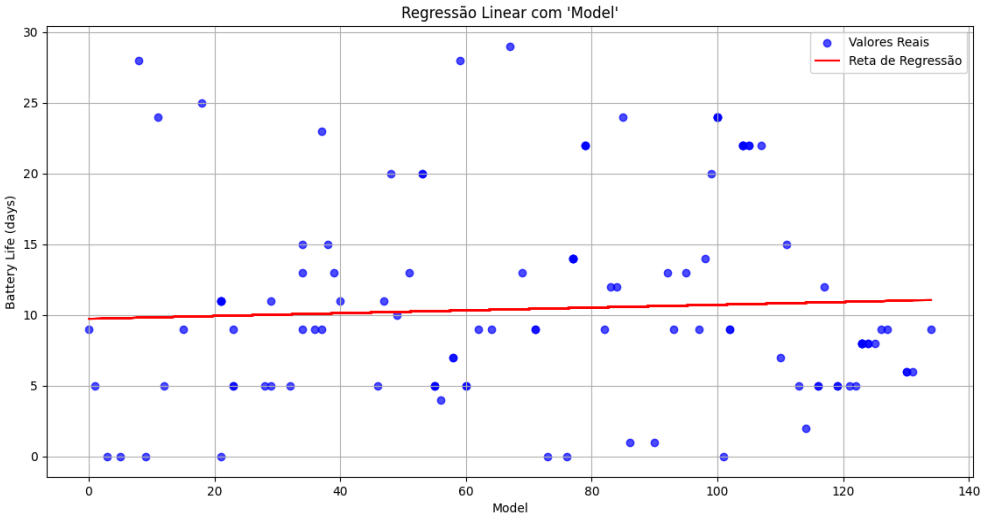


6) **Implemente uma regressão linear utilizando somente este atributo mais relevante, para predição do atributo alvo determinado na questão 5 também**

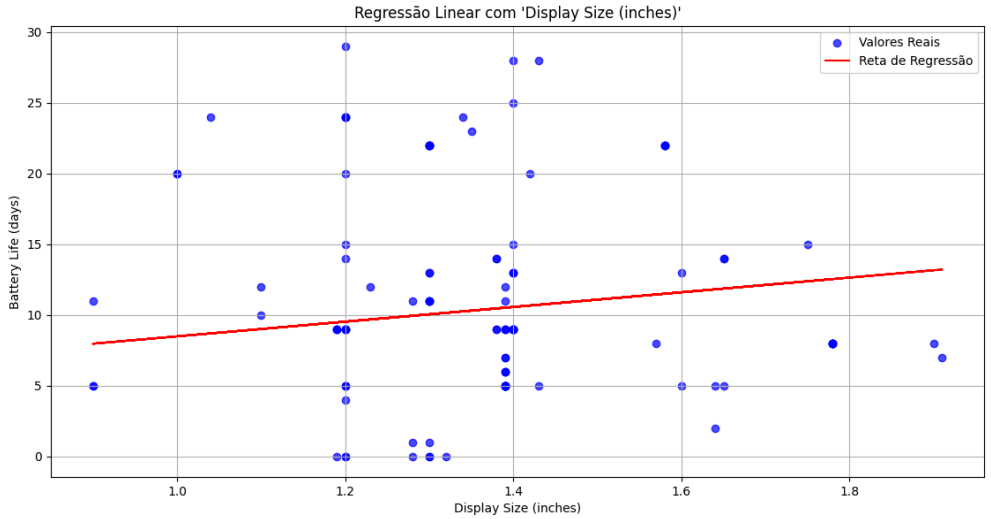
Como dito anteriormente, tentei fazer a regressão tendo como alvo a duração da bateria, o gráfico segue logo abaixo:



Nesse gráfico utilizei o preço como um atributo relevante para a duração da bateria, pensando que dispositivos comumente mais caros tivessem uma duração melhor em comparação com os mais baratos. Logo em seguida, pensei em usar o modelo e o SO do smartwatch, pois poderia consumir mais energia.



Display Size(inches)



7) Utilizando kfold e cross-validation faça uma regressão linear. Utilize uma implementação manual do kfold e cross-validation. calule as métricas RSS, MSE, RMSE e R_squared que também devem ser implementadas manualmente.

Para a implementação manual, ainda usando Battery Life (days) para a regressão, foi feita a separação dos dados de treino e teste. Criei uma função chamada k_fold_cross_validation para dividir os dados em várias partes e testar a precisão da previsão. Após aplicado o modelo fit de regressão linear. Cada uma das métricas foram implementadas manualmente de acordo com suas funções. Para cada "fold", guardei as métricas e, ao final, calculamos a média de todas as métricas. Nessa questão tive dificuldades principalmente quando se tratou do Kfold manual, não sei se está correta, mas este foi o resultado obtido:

4. Conclusões

Os resultados esperados foram satisfeitos? Se não, qual o motivo? Qual a sua análise?

Apesar de enfrentar muitas dificuldades, principalmente na implementação manual, tanto do KNN quando do Kfold, acredito que em parte, os resultados foram satisfatórios. Mas ainda preciso de prática, inclusive no aprofundamento na parte de regressão.

5. Próximos passos

Não sei o que sugerir no momento.