UNIVERSIDADE FEDERAL DO CEARÁ

TECHNICAL REPORT

Aluno: Helen Vitória Brandão de Sousa e Ingrid Melo de Oliveira

1. Introdução

Este relatório visa relatar sobre a elaboração da segunda avaliação da cadeira de inteligência artificial do curso de segurança da informação no campus Jardins de Anita, aplicando os conhecimentos obtidos durante as aulas presenciais. Neste iremos utilizar machine learning básico em Python, utilizando bibliotecas como pandas, numpy, scikit-learn, matplotlib.pyplot e suas funções.

Para a realização das questões do trabalho precisamos utilizar um dataset que foi disponibilizado pelo professor. O dataset é sobre reservas de hoteis, onde o objetivo é prever se o cliente vai honrar a reserva ou cancelá-la, levando em consideração fatores como mudança de planos ou conflitos de agendamento. O conjunto exige um processo de limpeza e recodificação dos dados para adequá-lo à análise. Logo abaixo, estão mais informações sobre o dataset.

Dataset de Classificação: Hotel Reservations Dataset

O dataset em questão contém dados sobre as reservas realizadas por clientes em hoteis, incluindo informações sobre a natureza dessas reservas e o comportamento dos clientes. Entre as variáveis registradas, estão informações sobre o cliente como idade dos clientes, gênero, massa corporal(IMC), quantidade de dependentes, região em que moram e suas rendas.

A análise desse conjunto de dados irá proporcionar a previsão da probabilidade de cancelamento de uma reserva, permitindo que a indústria hoteleira tenha insights valiosos, fazendo com que antecipem cancelamentos e adotem estratégias para minimizar o impacto desses eventos, consequentemente otimizando suas operações e as suas receitas.

Dicionário de Dados - Hotel Reservations Dataset:

- ***Booking_ID:*** *identificador único de cada reserva*
- ***no_of_adults:*** *Número de adultos*
- ***no_of_children:*** *Número de filhos*
- ***no_of_weekend_nights:*** *Número de noites de fim de semana (sábado ou domingo) em que o hóspede ficou ou reservou para ficar no hotel*
- ***no_of_week_nights:*** *Número de noites da semana (segunda a sexta) em que o hóspede ficou ou reservou para ficar no hotel*
- ***type_of_meal_plan:*** *Tipo de plano de refeições reservado pelo cliente:*
- ***required_car_parking_space:*** *O cliente precisa de uma vaga de estacionamento? (0 - Não, 1 - Sim)*
- ***room_type_reserved:*** *Tipo de quarto reservado pelo cliente. Os valores são cifrados (codificados) pelo INN Hotels.*
- ***lead_time:*** *Número de dias entre a data da reserva e a data de chegada*
- ***arrival_year:*** *Ano da data de chegada 2*
- ***arrival_month:*** *Mês da data de chegada*
- ***arrival_date:*** *Data do mês*
- ***market_segment_type:*** *Designação do segmento de mercado.*
- ***repeated_guest:*** *O cliente é um hóspede repetido? (0 - Não, 1 - Sim)*
- ***no_of_previous_cancellations:*** *Número de reservas anteriores que foram canceladas pelo cliente antes da reserva atual*
- ***no_of_previous_bookings_not_canceled:*** *Número de reservas anteriores não canceladas pelo cliente antes da reserva atual*
- ***avg_price_per_room:*** *Preço médio por dia da reserva; os preços dos quartos são dinâmicos. (em euros)*
- ***no_of_special_requests:*** *Número total de solicitações especiais feitas pelo cliente (por exemplo, andar alto, vista do quarto, etc.)*
- ***booking_status:*** *Indicador que indica se a reserva foi cancelada ou não.*

2. Observações

Na primeira questão, para garantir compatibilidade com a AV1, precisamos limitar o número de amostras para no máximo 5000, pois essa redução visa não sobrecarregar o processo de análise, para que possa ser possível realizar comparações precisas entre as avaliações.

Outro problema ocorrido neste trabalho, foi quando percebemos que logo após testar a métrica Mahalanobis na execução do GridSearchCV, percebemos que não funcionava corretamente. Para resolver esse problema, foi necessário criar uma matriz de covariância manualmente, o que corrigiu o erro e permitiu que o método funcionasse corretamente.

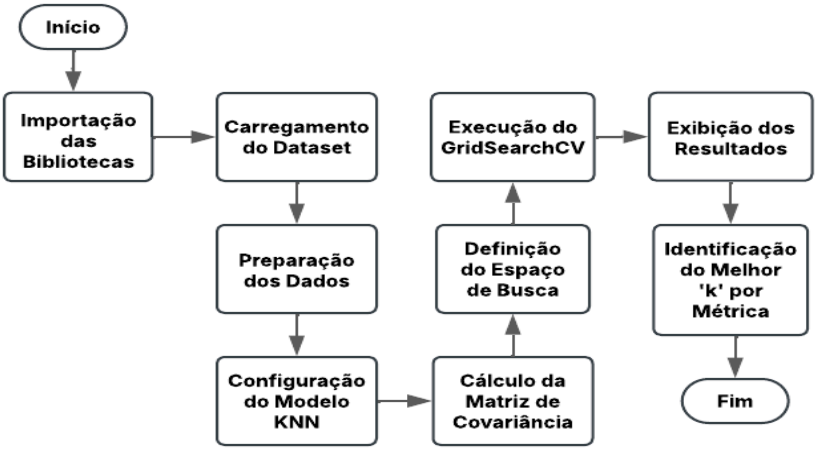
Além desses, nas questões 3 e 4, devido às limitações do recursos do computador, optamos por não utilizar os métodos do Cotovelo e Silhueta simultaneamente, pois o dataset que estamos usando contém um número significativo de dados. Portanto, para evitar sobrecarga no sistema, rodamos um método de cada vez e comentamos o código do método que não está sendo utilizado. Através dessa estratégia foi possível analisar cada método de forma individual, sem comprometer o desempenho da máquina.

3. Resultados e discussão

3.1 Primeira Questão

O objetivo desta questão é descobrir qual a melhor parametrização no processo de classificação do nosso dataset usando GridSearchCV e verificar se o resultado obtido na AV1 é o mesmo que obtivemos nesta AV2. Logo abaixo é apresentado o fluxograma e a explicação do mesmo sobre o processo da questão 01.

FLUXOGRAMA DO PROCESSO DA QUESTÃO 01



EXPLICAÇÃO DO FLUXOGRAMA

Conforme é ilustrado no fluxograma, o processo inicia com a importação das bibliotecas usadas durante a implementação, e logo após com o carregamento e preparação do dataset (dataset_classificacao_ajustado.csv). Em seguida os dados foram divididos em treino (70%) e teste (30%) com o train_test_split, e reduzidos para 5000 amostras para manter comparação com a AV1.

Na parte seguinte, um classificador (KNN), é inicializado e validado com o Kfold em 5 divisões. A métrica Mahalanobis precisou de uma matriz de covariância estimada, através de (EmpiricalCovariance) e sua inversa (IV) foi calculada.

Logo depois, é feita a busca por melhores hiperparâmetros que ocorre via GridSearchCV, ele testa diferentes combinações de k e métricas (chebyshev, manhattan, euclidean, mahalanobis). Os melhores parâmetros são extraídos com (knn_cv.best_ params_), e a melhor acurácia obtida é registrada. Os resultados finais destacam o melhor k para cada métrica e a melhor parametrização.

3.1.1 Resultados

Os resultados obtidos foram satisfatórios, pois conseguimos identificar a melhor parametrização no processo de classificação do nosso dataset, identificamos que o melhor k foi 7 e a melhor métrica foi Mahalanobis que obteve uma acurácia de 0.8526.

Após a obtenção dos resultados da segunda avaliação, fizemos a comparação com os resultados da primeira avaliação e analisamos que ambos os resultados deram mahalanobis como melhor métrica, porém com apenas uma pequena diferença nas

acurácias. Pois na primeira avaliação, a métrica Mahalanobis teve uma acurácia de 0.8349 enquanto na segunda avaliação de 0.852631.

Logo abaixo é apresentado as tabelas dos resultados obtidos:

TABELA DE MELHOR PARAMETRIZAÇÃO DA AV2

param_metric	param_n_neighbors	mean_test_score
chebyshev	3	0.797496
euclidean	3	0.804506
mahalanobis	7	0.852631
manhattan	3	0.814036

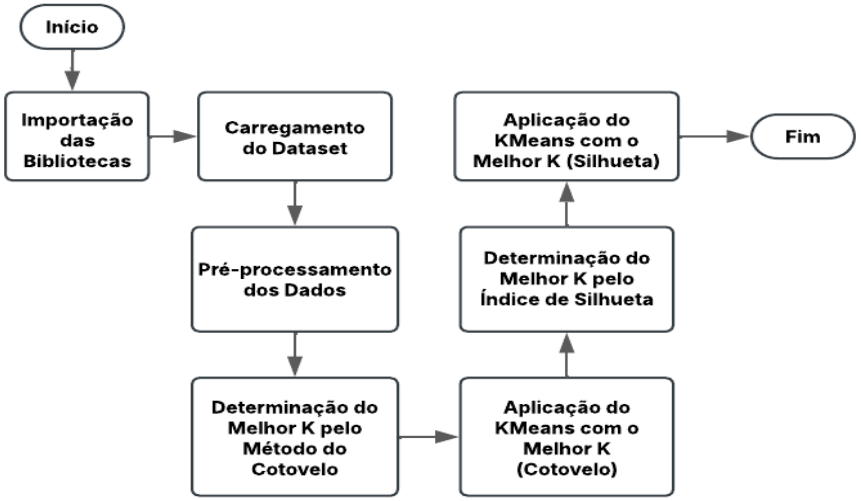
TABELA DE MELHOR PARAMETRIZAÇÃO DA AV1

param_metric	param_n_neighbors	mean_test_score
chebyshev	7	0.7756
euclidean	7	0.7767
mahalanobis	7	0.8349
manhattan	7	0.7842

3.2 Segunda Questão

O objetivo desta questão é fazer uma análise para descobrir a quantidade ideal de clusters que deve ser feita, desconsiderando a coluna alvo do dataset e considerando utilizar o método cotovelo e silhueta. Logo abaixo é apresentado o fluxograma e a explicação do mesmo sobre o processo da questão 02.

FLUXOGRAMA DO PROCESSO DA QUESTÃO 02



EXPLICAÇÃO DO FLUXOGRAMA

O início do fluxograma ilustra o processo inicial com a importação das bibliotecas necessárias, e segue com o carregamento e pré-processamento dos dados (dataset_classificacao_ajustado.csv). A variável alvo (booking_status) é removida, e os dados são normalizados com o StandardScaler, para evitar que o K-Means sofra influência de escalas.

Depois, o número ideal de clusters (K) é determinado por dois métodos, o método do cotovelo onde o K-Means é treinado para K de 1 a 10, registrando a inércia e pelo índice de silhueta, onde o K-Means é ajustado para K de 2 a 10. O coeficiente de silhueta determina o K com melhor separação entre clusters.

Por fim, são gerados gráficos para que seja feita a análise e definição do melhor K, considerando a estabilização da inércia e a separação dos clusters.

3.2.1 Resultados

Os resultados obtidos foram satisfatórios, pois através da análise dos gráficos conseguimos identificar a quantidade ideal de clusters que deve ser feita para o nosso dataset. No método do Cotovelo obtivemos (K=4) que apresenta um "bom valor" comparado a outros pontos e no método de Silhueta o maior valor de Silhouette Score ocorre em (K=9), o que indica que esses valores oferecem a melhor separação entre os clusters.

Logo abaixo é apresentado os gráficos dos métodos: Cotovelo e Silhueta

GRÁFICO DO MÉTODO DO COTOVELO

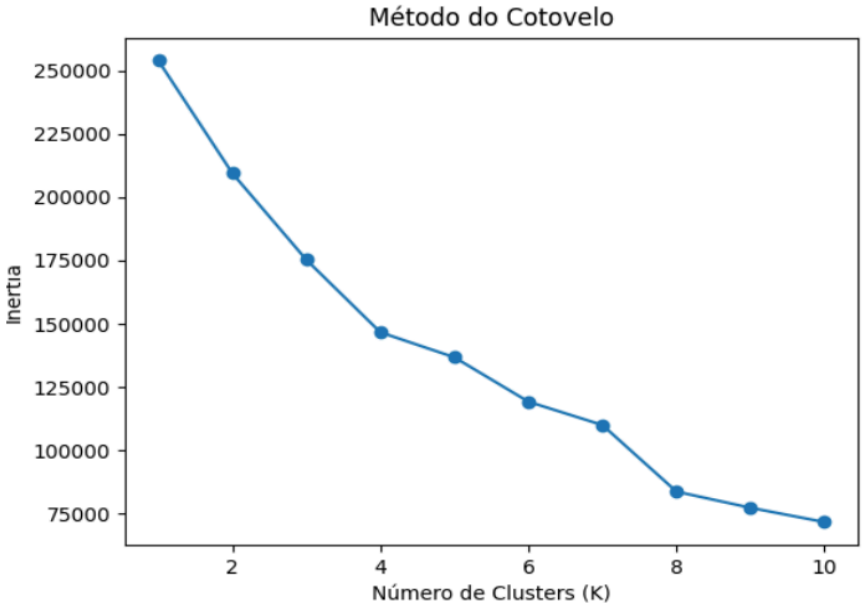


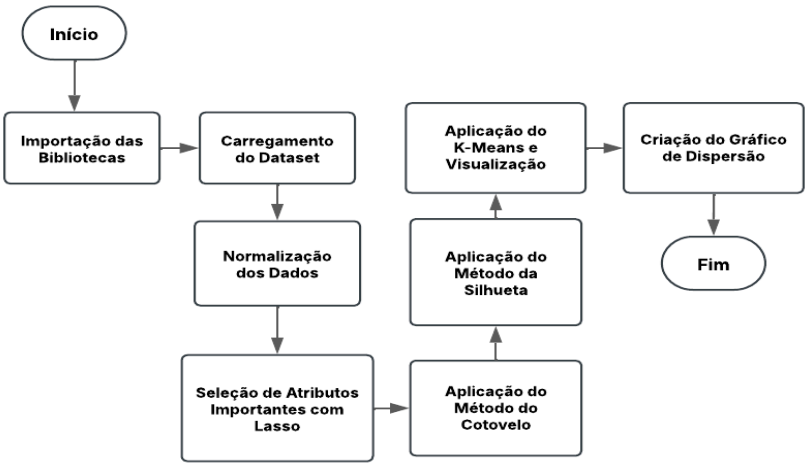
GRÁFICO DE ÍNDICE DE SILHUETA

3.3 Terceira Questão

O objetivo desta questão é refazer os métodos cotovelo e silhueta utilizando os atributos mais relevantes do dataset escolhidos pelo método lasso, caso mude a

quantidade de clusters com relação à questão anterior fazer dois scatterplots e uma análise visual. Logo abaixo é apresentado o fluxograma e a explicação do mesmo sobre o processo da questão 03.

FLUXOGRAMA DO PROCESSO DA QUESTÃO 03



EXPLICAÇÃO DO FLUXOGRAMA

O processo inicia com importação das bibliotecas necessárias e o pré-processamento dos dados, onde o dataset (dataset_classifcacao_ajustado.csv) é carregado e a variável alvo (booking_status) é separada. Depois, os dados são normalizados com o StandardScaler para garantir escalas padronizadas, para que a clusterização com K-Means seja melhor. Após a normalização dos dados é feita a seleção de atributos com o Lasso, que identifica as variáveis mais relevantes.

Para definir o número ideal de clusters, dois métodos são aplicados. O Método do Cotovelo executa o K-Means para K variando de 1 a 10, registrando a inércia e identificando o melhor K pelo ponto de inflexão no gráfico. Enquanto o Método da Silhueta calcula o coeficiente de silhueta para K entre 2 e 10, escolhendo o valor com melhor separação de clusters.

Por fim, o K-Means é aplicado com os valores de K escolhidos, e os clusters formados são visualizados em gráficos de dispersão, destacando a distribuição dos grupos e seus centroides.

3.3.1 Resultados

Ao obtermos os novos valores para o método cotovelo e silhueta desta, usando o método Lasso (Least Absolute Shrinkage and Selection Operator) que escolhe automaticamente as variáveis mais relevantes do nosso dataset para evitar overfitting, que no nosso caso foram ['lead_time', 'no_of_special_requests']. Foi percebido que, com relação à questão anterior o número de k mudou significativamente, pois na segunda questão o método cotovelo deu (k=4), enquanto na terceira questão deu como resultado (k=2), já o método silhueta da segunda questão deu (k=9), enquanto na terceira questão foi (k=2).

Por fim, a questão exigia que se houvesse mudança no número de cluster (k), que foi o que aconteceu, fosse realizado uma análise visual com dois scatterplots. Nós o fizemos, e logo abaixo é apresentado as tabelas e os scatterplots dos resultados obtidos:

GRÁFICO DO MÉTODO DO COTOVELO

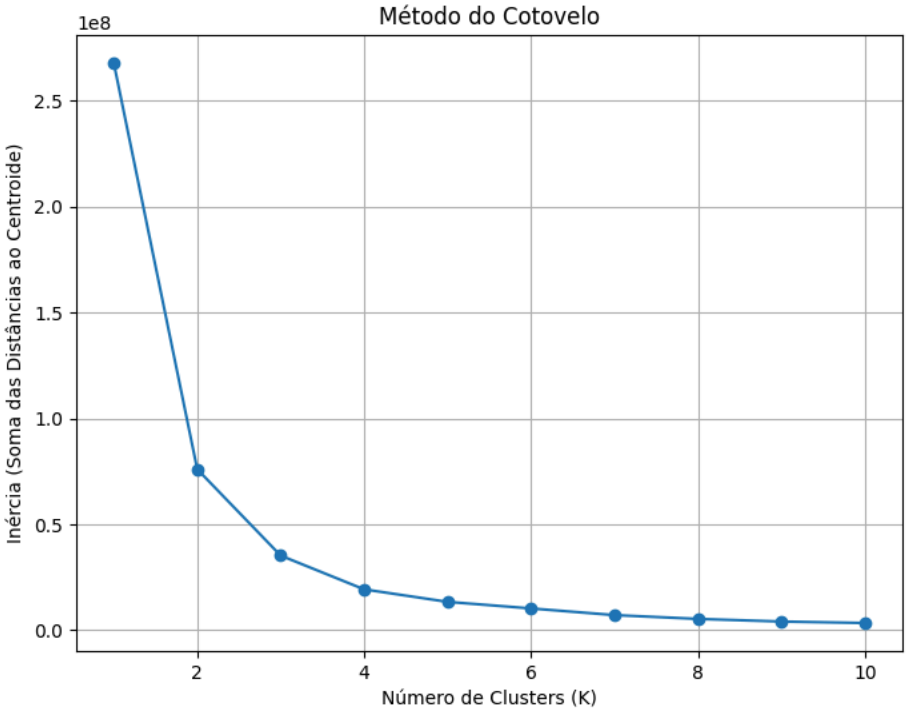


GRÁFICO DO MÉTODO DO SILHUETA

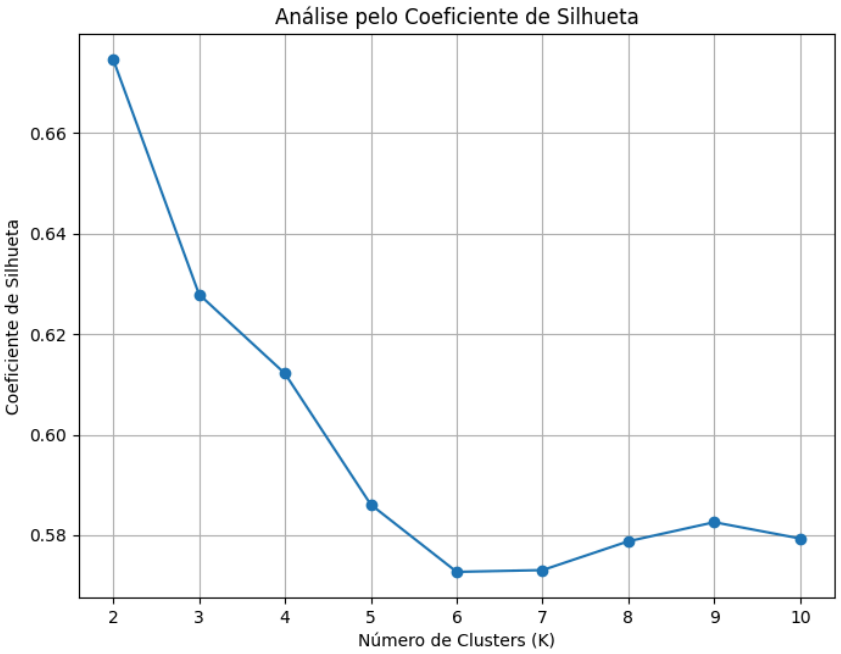


GRÁFICO DE CLUSTER DO MÉTODO DO COTOVELO

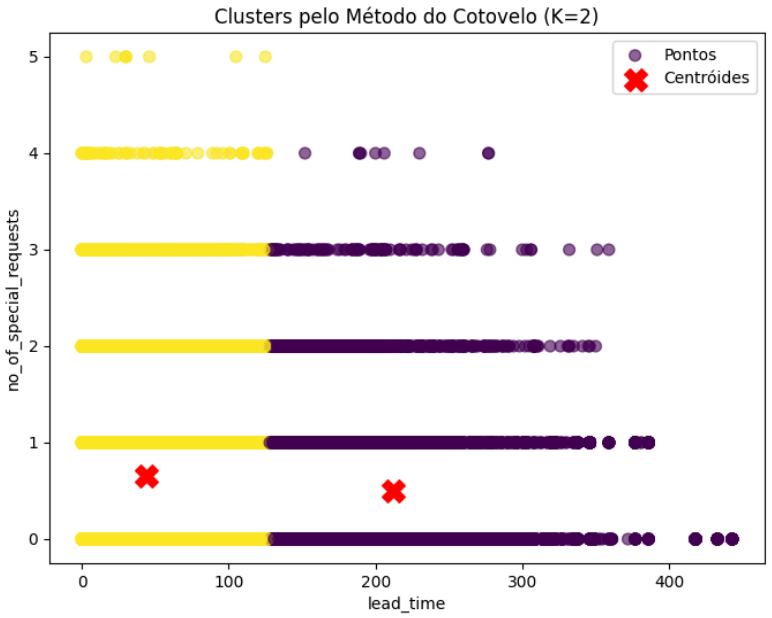
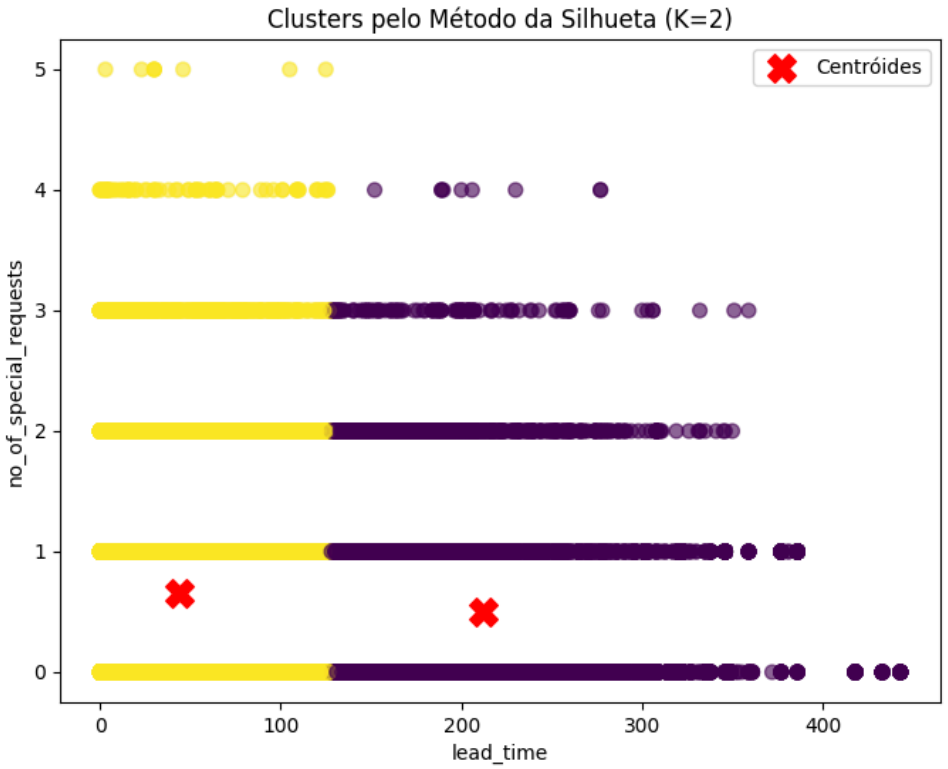


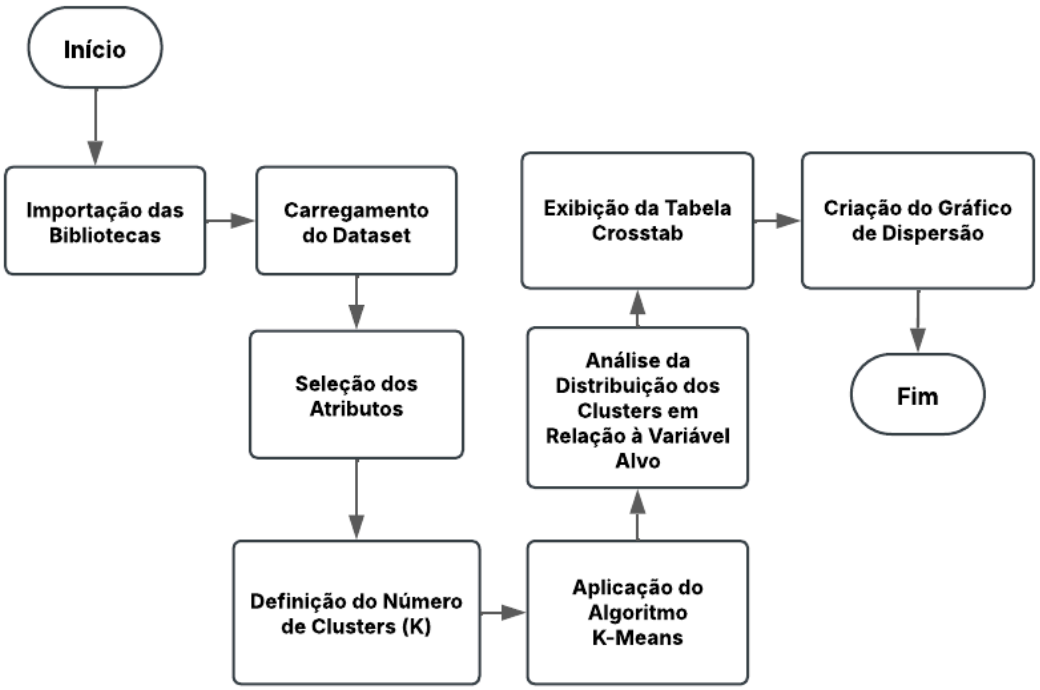
GRÁFICO DE CLUSTER DO MÉTODO DO SILHUETA



3.4 Quarta Questão

O objetivo desta questão é fazer um crosstab para que possamos identificar como ficou a distribuição de cluster de acordo com classes presentes na coluna alvo do seu dataset, utilizando o KMeans e o k obtido pelo índice de silhueta. Logo abaixo é apresentado o fluxograma e a explicação do mesmo sobre o processo da questão 04.

FLUXOGRAMA DO PROCESSO DA QUESTÃO 04



EXPLICAÇÃO DO FLUXOGRAMA

O fluxograma inicia com a importação das bibliotecas para realizar as implementações e depois segue com o carregamento do dataset. Em seguida, faz a seleção de atributos mais relevantes, que foram lead_time e no_of_special_requests.

Na próxima etapa, ocorre a definição do número de clusters (K), o melhor valor obtido baseado no Índice de Silhueta, que foi K=2. Com essa informação, o modelo K-Means é treinado, atribuindo clusters às amostras. Em seguida ocorre a análise da relação entre os clusters e a variável alvo, que envolve a criação de um dataframe auxiliar com os clusters e booking_status, seguido da geração de uma tabela cruzada (crosstab) para verificar a distribuição.

Por fim, na visualização gráfica, é criado um gráfico de dispersão com pontos coloridos conforme seus clusters, onde o eixo X representa lead_time e o Y representa no_of_special_requests, com um colorbar para identificação dos clusters. Essa sequência, representada no fluxograma, oferece uma análise clara dos dados e da formação dos clusters.

3.4.1 Resultados

Como resultado, ao obtermos o crosstab utilizando o KMean e o k óbitos pelo índice de silhueta, identificamos que a distribuição de clusters ficou da seguinte forma: cluster 0 e cluster 1. E como classes também obtivemos duas: booking_status = 0 (reservas canceladas) e booking_status = 1 (reservas não canceladas).

Ao analisarmos a tabela fornecida pelo crosstab, percebemos a relação entre os clusters e as classes. No cluster 0 obtivemos 5791 reservas canceladas e 3194 reservas confirmadas, já no cluster 1 tivemos 6094 reservas canceladas e 21196 reservas confirmadas.

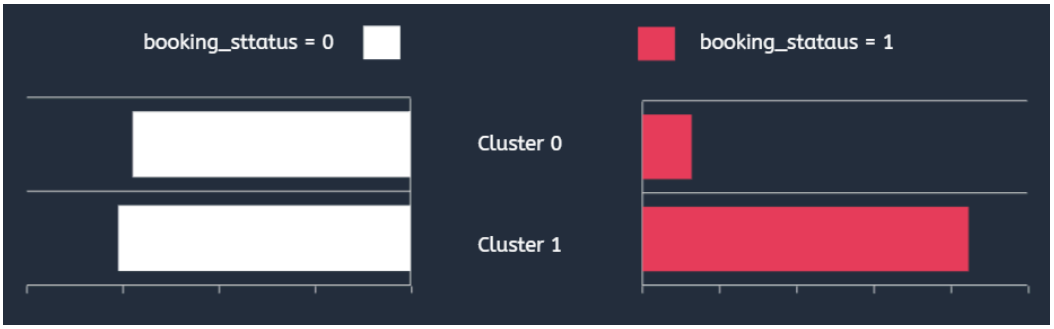
Portanto, é possível identificar que no cluster 0 as reservas canceladas se sobressaem sobre as reservas confirmadas, já no cluster 1 ocorre o contrário, pois as reservas confirmadas são maiores que as canceladas.

Logo abaixo é apresentado a tabela obtida do crosstab, um gráfico em barras e um scatterplots para melhor visualização da diferença de resultado:

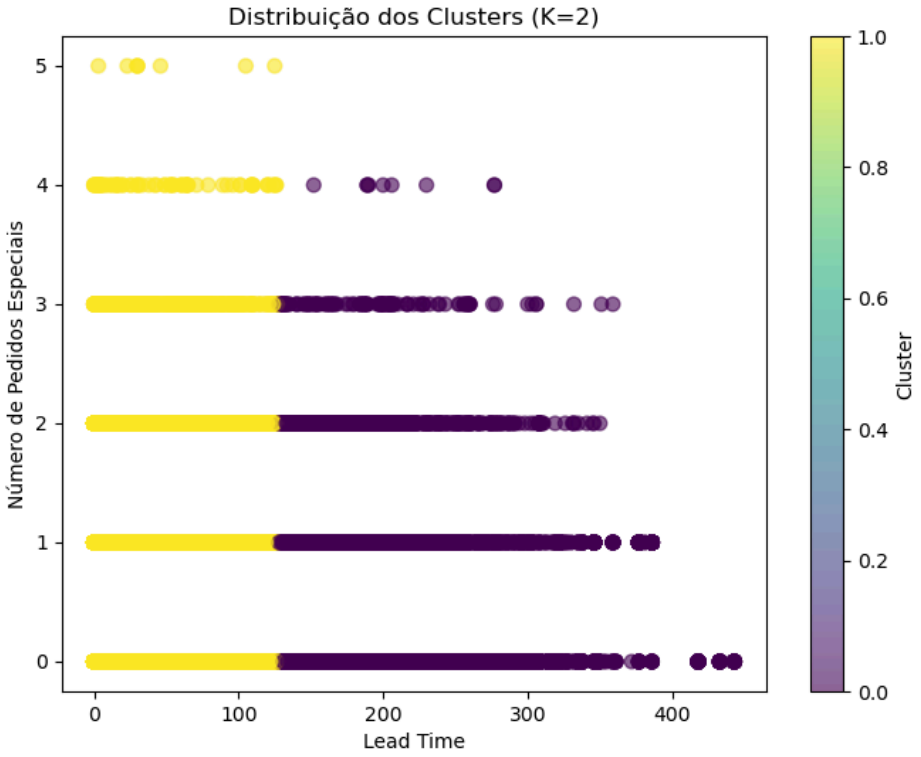
TABELA DO RESULTADO DO CROSSTAB

Cluster	booking_status = 0	booking_status = 1	Total
0	5791	3194	**8985**
1	6094	21196	**27290**
Total	**11885**	**24390**	**36275**

GRÁFICO EM BARRAS DO RESULTADO DO CROSSTAB



SCATTERPLOT DO RESULTADO DO CROSSTAB



4. Conclusões

Os resultados obtidos foram satisfatórios, permitindo identificar padrões importantes no dataset e aprimorar a tomada de decisão.

Na Questão 1, conseguimos definir a melhor parametrização para o processo de classificação. O valor ideal de k foi 7, e a métrica Mahalanobis apresentou a melhor acurácia, atingindo 0.8526. Comparando com a primeira avaliação, observamos que Mahalanobis continuou sendo a melhor métrica, mas com um leve aumento na acurácia em relação ao primeiro teste (0.8349 para 0.8526).

Na Questão 2, analisamos a melhor quantidade de clusters para o dataset. Utilizando os métodos do Cotovelo e da Silhueta, identificamos que K=4 foi o melhor valor no método do Cotovelo, enquanto o maior Silhouette Score ocorreu em K=9, indicando que esses valores oferecem a melhor separação entre os clusters.

Na Questão 3, aplicamos o método Lasso (Least Absolute Shrinkage and Selection Operator) para selecionar automaticamente as variáveis mais relevantes e

evitar overfitting. Isso impactou diretamente a determinação do número ideal de clusters. Comparando com os valores obtidos na questão anterior, notamos uma mudança significativa: no método do Cotovelo, o K passou de 4 para 2, e no método da Silhueta, de 9 para 2. Essa alteração demonstra como a escolha das variáveis influencia na estruturação dos clusters.

Na Questão 4, realizamos uma análise visual dos clusters utilizando scatterplots e analisamos a relação entre clusters e classes através do crosstab. Os resultados mostraram que no cluster 0, houve mais reservas canceladas (5791 canceladas vs. 3194 confirmadas), enquanto no cluster 1, a maioria das reservas foi confirmada (21196 confirmadas vs. 6094 canceladas). Isso evidencia um padrão interessante: no cluster 0, os cancelamentos superam as confirmações, enquanto no cluster 1, ocorre o contrário.

Portanto, concluímos que os métodos aplicados foram eficazes na classificação e análise dos dados, permitindo obter insights valiosos sobre o comportamento das reservas.

5. Próximos passos

Após a análise do relatório e da apresentação, recomendamos que para uma melhor aplicação do projeto, sejam feitos os seguintes passos:

***Transformação de Dados Categóricos**: O nosso dataset possui muitas colunas com dados categóricos, o que pode ter influenciado na separação dos clusters e na qualidade das visualizações, pois os gráficos scatterplots mostraram os clusters muito próximos uns dos outros. Então sugerimos que seja feita uma conversão de dados categóricos para numéricos, isso pode melhorar a qualidade das visualizações.*

***Utilização do Classification Report**: Após a transformação dos dados, você pode aplicar o classification report para avaliar a performance do modelo. Isso permitirá uma análise mais detalhada da precisão, recall e F1-score dos clusters identificados, contribuindo para a melhoria contínua do modelo.*