Análise de Dados do *Khiata*

TMJ (Transformando o Mundo Juntos)

Enzo – N°2

Inaldo – N°10

Sumário

[Introdução 3](#_Toc181716025)

[Objetivo do aplicativo 3](#_Toc181716026)

[Objetivo da Análise Exploratória de Dados 3](#_Toc181716027)

[Objetivo da Análise de Dados (para Aprendizado de Máquina) 4](#_Toc181716028)

[Levantamento dos dados na análise exploratória 4](#_Toc181716029)

[Busca dos dados 4](#_Toc181716030)

[Justificativa de uso 4](#_Toc181716031)

[Descrição da base de dados de trabalho 5](#_Toc181716032)

[Limpeza dos dados 5](#_Toc181716033)

[Condicionamento para alimentar o modelo de ML 5](#_Toc181716034)

[Condicionamento inicial 6](#_Toc181716035)

[ANÁLISE DE DADOS 6](#_Toc181716036)

[Definição dos objetivos e das classes 6](#_Toc181716037)

[Definição dos modelos mais adequados para analisar os dados 6](#_Toc181716038)

[Descrição dos modelos selecionados 7](#_Toc181716039)

[Decision Tree: 7](#_Toc181716040)

[Ajuste: 7](#_Toc181716041)

[Precisão e Robustez: 7](#_Toc181716042)

[Interpretação e Explicabilidade: 7](#_Toc181716043)

[Desempenho Computacional: 7](#_Toc181716044)

[Capacidade de Generalização: 7](#_Toc181716045)

[Ajuste: 7](#_Toc181716046)

[Precisão e Robustez: 7](#_Toc181716047)

[Desempenho Computacional: 8](#_Toc181716048)

[Capacidade de Generalização: 8](#_Toc181716049)

[Ajuste: 8](#_Toc181716050)

[Precisão e Robustez: 8](#_Toc181716051)

[Interpretação e Explicabilidade: 8](#_Toc181716052)

[Desempenho Computacional: 8](#_Toc181716053)

[Capacidade de Generalização: 8](#_Toc181716054)

[Aplicação dos Modelos Selecionados 8](#_Toc181716055)

[Análise dos Resultados 9](#_Toc181716056)

[Comparação Entre Modelos 10](#_Toc181716057)

[Comparação Direta: 10](#_Toc181716058)

[Forças e Fraquezas: 10](#_Toc181716059)

[Ajustes Necessários 10](#_Toc181716060)

[Pré-processamento: 10](#_Toc181716061)

[Ajustes nos Modelos: 10](#_Toc181716062)

[Impacto das Modificações 10](#_Toc181716063)

[Conclusão 11](#_Toc181716064)

[Integração da IA no aplicativo 13](#_Toc181716065)

[RPA para o banco 13](#_Toc181716066)

# Introdução

Muitas costureiras independentes enfrentam desafios para expandir seu alcance e alcançar clientes que valorizam o trabalho artesanal e personalizado. A concorrência com grandes plataformas e a dificuldade de criar uma presença digital sólida podem ser barreiras significativas. Além disso, para clientes que buscam produtos exclusivos e personalizados, pode ser complicado encontrar artesãs confiáveis e se comunicar diretamente com elas para ajustar detalhes específicos.

O Khiata, desenvolvido pela TMJ, é uma plataforma voltada para conectar costureiras a clientes que valorizam moda e artesanato sob medida. Nosso público-alvo inclui aqueles que procuram produtos exclusivos e personalizados, além de costureiras que desejam expandir sua clientela e melhorar sua presença digital. Através do app, os usuários podem conversar diretamente com as costureiras, explorar o portfólio de produtos e encomendar peças de acordo com suas preferências, promovendo uma experiência de compra acessível e personalizada.

## Objetivo do aplicativo

O aplicativo Khiata conecta clientes a costureiras independentes em todo o Brasil, facilitando a compra de produtos artesanais e personalizados. Nosso objetivo é valorizar o trabalho artesanal, tornando a moda sob medida e a produção sustentável mais acessíveis a todos. No Khiata, os usuários podem encontrar costureiras que compartilhem seus valores e estilo, conversar diretamente com elas para ajustar detalhes de suas encomendas, e adquirir produtos que atendam exatamente às suas preferências e necessidades.

O app funciona como uma vitrine digital para as costureiras, permitindo que elas publiquem seus produtos e se conectem com clientes que buscam peças únicas e sustentáveis. Além disso, o Khiata sugere "matches" entre clientes e costureiras, com base em preferências, estilo e necessidades, tornando o processo de compra e comunicação mais ágil e eficiente.

# Objetivo da Análise Exploratória de Dados

Montar, analisar e limpar um conjunto de dados relacionado ao aplicativo para ser utilizado em análise de dados exploratória e também para fins de aprendizado de máquina (ML).

O objetivo da análise exploratória é melhorar a compreensão dos dados do conjunto sobre os usuários potenciais do aplicativo, encontrando padrões, tendências e discrepâncias possíveis. Essa etapa é crucial para identificar os recursos mais importantes na previsão do interesse do aplicativo e para otimizar os dados, melhorando a precisão de futuros modelos de aprendizado de máquina.

Nesta etapa, examinamos a distribuição dos dados e realizamos transformações nas variáveis categóricas para facilitar a interpretação dos modelos de machine learning. Também identificamos a necessidade de balancear as classes para garantir um treinamento mais justo. O pré-processamento das variáveis categóricas e o balanceamento das classes com SMOTE foram essenciais para alcançar um desempenho mais preciso e imparcial dos modelos.

Em resumo, a análise exploratória proporcionou uma visão inicial do comportamento dos potenciais usuários, contribuindo para aprimorar tanto a estratégia de segmentação quanto os algoritmos de previsão a serem implementados.

# Objetivo da Análise de Dados (para Aprendizado de Máquina)

A análise de dados no Khiata visa identificar as variáveis que mais influenciam o interesse dos usuários pela plataforma, destacando os fatores que motivam alguém a explorar e utilizar o aplicativo. Buscamos entender quais características têm maior impacto na decisão, como idade, preferências de estilo, preocupação com sustentabilidade e familiaridade com compras de produtos artesanais. Esse processo envolve o processamento e a análise detalhada dos dados dos usuários e das costureiras.

Essas percepções são fundamentais para ajustar e aprimorar os modelos de aprendizado de máquina, garantindo a precisão e a confiabilidade das recomendações de produtos e das sugestões de "matches" entre clientes e costureiras. Além disso, a análise permite uma segmentação mais eficiente do público e auxilia no planejamento de estratégias específicas para atrair novos usuários e apoiar costureiras em sua jornada de crescimento no aplicativo.

# Levantamento dos dados na análise exploratória

## Busca dos dados

As informações utilizadas foram coletadas por meio de formulários distribuídos em grupos de Facebook de costureiras, capturando dados reais de possíveis usuários do aplicativo. A base de dados contém características como faixa etária, frequência de consumo de produtos artesanais, interesse em tipos específicos de produtos e hábitos de uso de aplicativos relacionados. Esses elementos foram estrategicamente escolhidos para oferecer uma visão detalhada dos fatores que podem impactar o interesse no aplicativo, como o nível de interesse em produtos artesanais, a faixa de preço preferida e a frequência de consumo. A análise visa identificar tendências e segmentações de perfil com maior probabilidade de adoção do aplicativo, possibilitando ajustes em estratégias de marketing e melhorias de funcionalidades conforme o perfil dos usuários potenciais.

## Justificativa de uso

A seleção dos dados é fundamental para obter uma visão completa dos possíveis usuários do aplicativo Khiata, uma vez que cada atributo escolhido representa aspectos importantes que influenciam sua decisão de uso. Por exemplo, características como faixa etária e preferências de estilo ajudam a identificar o perfil dos usuários, enquanto fatores como interesse em produtos sustentáveis e familiaridade com plataformas de compra online oferecem insights sobre a probabilidade de engajamento com o aplicativo. Além disso, a confiança nas avaliações de outros clientes destaca o papel das recomendações na tomada de decisão.

Esses dados não apenas guiaram o desenvolvimento de estratégias para aumentar a adoção do aplicativo, mas também aprimoraram a precisão dos modelos de aprendizado de máquina, garantindo que a seleção seja fundamentada em informações altamente relevantes. Destacar essas justificativas em uma seção específica permite ao leitor entender o valor de cada característica e sua contribuição para o objetivo principal do projeto.

# Descrição da base de dados de trabalho

A base de dados é composta por variáveis categóricas e numéricas, como faixa etária, frequência de compra de produtos artesanais, tipo de produto mais interessante e faixa de preço de interesse, todas essenciais para capturar os diferentes perfis dos usuários em potencial. Para utilizar essas variáveis categóricas nos modelos de machine learning, aplicamos o Label Encoding, convertendo-as em valores numéricos, mantendo o significado das categorias.

A coluna alvo, "Utilizaria\_App\_Costura", que indica a intenção dos usuários de adotar o aplicativo, apresentava um desbalanceamento, com uma predominância de uma das classes. Para corrigir isso, utilizamos a técnica SMOTE, que cria novas instâncias sintéticas da classe minoritária, equilibrando a base e favorecendo uma análise mais precisa e confiável dos modelos.

Para avaliar o desempenho dos modelos, dividimos a base em treino e teste na proporção de 80/20. Essa divisão proporciona uma amostra ampla para o treino e uma quantidade adequada de dados para testar a eficácia dos modelos, evitando o overfitting e permitindo uma avaliação robusta dos resultados.

# Limpeza dos dados

O processo de preparação dos dados foi fundamental para a análise e o treinamento dos modelos. Primeiro, as variáveis categóricas, como "faixa etária" e "tipo de produto de maior interesse", foram convertidas em valores numéricos usando OrdinalEncoder e LabelEncoder. Dessa forma, mantivemos o significado das categorias enquanto permitimos que o modelo interpretasse esses dados de forma consistente, possibilitando comparações eficazes entre diferentes perfis.

Também foi essencial equilibrar as classes da variável alvo, "Utilizaria\_App\_Costura". Esta coluna estava desbalanceada, com uma maioria de respostas tendendo para uma das opções ("Sim" ou "Não"). Para lidar com isso, aplicamos o SMOTE, gerando exemplos sintéticos da classe minoritária. Esse procedimento ajuda o modelo a aprender igualmente sobre ambas as classes, evitando que ele fique "viciado" em prever a classe majoritária, o que poderia prejudicar a precisão e a capacidade de generalizar.

# Condicionamento para alimentar o modelo de ML

Os dados passaram pelas seguintes etapas de preparação antes de serem aplicados aos modelos de machine learning:

Transformação de Variáveis: As variáveis categóricas, como perfil de interesse e frequência de compra, foram convertidas em valores numéricos usando Label Encoding, permitindo que o modelo interpretasse essas informações de maneira adequada.

Balanceamento de Classes: Aplicamos o SMOTE para evitar um viés em relação à classe majoritária na variável "Utilizaria\_App\_Costura", garantindo que o modelo tivesse aprendizado equilibrado entre usuários que utilizariam e não utilizariam o aplicativo.

Normalização/Escalonamento: As variáveis numéricas foram normalizadas para uma escala comum, de modo que nenhuma característica tivesse peso excessivo, assegurando que todas as variáveis contribuam de maneira equilibrada para o desempenho do modelo.

## Condicionamento inicial

Redução de Dimensionalidade: Foi aplicado o método de Análise de Componentes Principais (PCA) para reduzir a dimensionalidade dos dados. Esse processo simplifica o modelo ao preservar variáveis essenciais e remover variáveis redundantes, o que melhora a eficiência do modelo sem perda significativa de informações relevantes sobre os interesses e comportamentos dos usuários.

Seleção de Variáveis com Alta Correlação: Com base nos dados analisados, foi identificada a correlação mais alta entre variáveis, o que permite identificar padrões sutis, como a relação entre preferências do usuário e seu interesse em moda sob medida. Isso auxilia na criação de "matches" mais precisos entre clientes e costureiras.

# ANÁLISE DE DADOS

# Definição dos objetivos e das classes

Previsão de Adesão ao Uso do Aplicativo

Previsão de Adesão ao App: Avaliar a probabilidade de um cliente usar o app Khiata para adquirir produtos artesanais e personalizados.

Identificação de Fatores de Influência: Avaliar quais variáveis, como idade e interesse por produtos artesanais, impactam mais a decisão de uso do app.

Classes da Resposta (y):

Classe 1: Sim - Clientes que afirmam que usariam o app, sendo um grupo potencial para campanhas de engajamento.

Classe 2: Não - Clientes que indicam não querer usar o app, permitindo o desenvolvimento de estratégias para aumentar seu interesse.

# Definição dos modelos mais adequados para analisar os dados

Com base nas características dos dados do Khiata, foram selecionados quatro modelos de machine learning para análise e comparação, cada um escolhido por suas qualidades de interpretação, eficiência e capacidade de segmentação:

Árvore de Decisão: Este modelo é eficaz para interpretar decisões complexas e identificar padrões significativos. Sua estrutura hierárquica permite uma fácil interpretação, o que o torna especialmente útil para categorizar usuários do Khiata com base em suas preferências, perfil de consumo e disposição em consumir moda artesanal. Ao utilizar o critério Gini ajustado, o modelo foi capaz de equilibrar as previsões entre as classes “Sim” e “Não”, alcançando uma acurácia média de 0.65 conforme a validação cruzada e mostrando menor tendência a enviesamentos em relação a outros modelos, como o Naive Bayes.

Naive Bayes: Este é um modelo eficiente e bem adequado para dados categóricos, como aqueles usados para avaliar o interesse por moda artesanal e a propensão de uso do app Khiata. Ele é ideal para identificar potenciais usuários, já que faz suposições independentes que simplificam o processo de classificação. No entanto, apresentou um certo enviesamento ao favorecer uma das classes, o que pode limitar a imparcialidade das previsões.

K-Nearest Neighbors (KNN): Este modelo é eficaz para segmentar perfis similares de clientes, permitindo que o Khiata recomende costureiras com base no estilo e nas preferências do cliente. A proximidade entre os dados torna o KNN ideal para capturar sutilezas nos gostos dos usuários e oferecer sugestões personalizadas. No entanto, sua aplicabilidade pode ser limitada para grandes volumes de dados, onde a computação de distâncias pode se tornar custosa.

# Descrição dos modelos selecionados

Cada modelo foi ajustado com hiperparâmetros específicos usando GridSearchCV para otimizar o desempenho nas previsões de adesão e preferências dos usuários no Khiata:

## Decision Tree:

Ajuste: O modelo foi otimizado com diferentes critérios de divisão, como Gini e Entropia, e variadas profundidades, o que melhorou a precisão e reduziu o risco de overfitting.

Precisão e Robustez: O Decision Tree demonstrou boa capacidade de classificar corretamente os dados, mantendo um equilíbrio entre precisão e recall, especialmente para prever o interesse dos usuários em produtos personalizados.

Interpretação e Explicabilidade: A estrutura em árvore facilita a visualização das decisões, tornando o modelo altamente interpretável, o que ajuda a compreender como variáveis como faixa etária e interesse em artesanato impactam a decisão de uso do app.

Desempenho Computacional: Relativamente eficiente, o modelo não requer muitos recursos para treinar e aplicar, mantendo bom desempenho mesmo com um conjunto de dados extenso.

Capacidade de Generalização: O ajuste de hiperparâmetros melhorou a generalização, permitindo que o modelo mantenha um desempenho consistente em novos dados, como os de futuros usuários.

K-Nearest Neighbors (KNN):

Ajuste: Foram testados diferentes números de vizinhos próximos para avaliar o impacto nas classificações, ajudando a definir a quantidade ideal para segmentar perfis semelhantes no Khiata.

Precisão e Robustez: KNN apresentou boa precisão ao classificar usuários com perfis semelhantes, especialmente útil para recomendar costureiras com estilos que combinam com o do cliente.

Interpretação e Explicabilidade: Embora intuitivo para entender a lógica de "proximidade", o KNN pode ser menos interpretável quando aplicado a dados de alta dimensionalidade.

Desempenho Computacional: KNN pode se tornar custoso para grandes volumes de dados, pois precisa calcular distâncias para cada previsão; portanto, é mais adequado para conjuntos de dados menores ou segmentados.

Capacidade de Generalização: A generalização do KNN é limitada para dados altamente variáveis; no entanto, com ajustes adequados, ele é eficaz para previsões personalizadas no app.

Naive Bayes:

Ajuste: Selecionado como modelo base, o Naive Bayes não exigiu muitos ajustes, sendo rápido e eficiente para lidar com variáveis categóricas do Khiata.

Precisão e Robustez: Apresentou desempenho consistente em dados categóricos e uma precisão aceitável para prever usuários potenciais. No entanto, o modelo mostrou certa tendência a enviesamento devido à suposição de independência entre as variáveis.

Interpretação e Explicabilidade: Extremamente interpretável, o Naive Bayes facilita a compreensão das previsões com base em probabilidades explícitas.

Desempenho Computacional: O modelo é altamente eficiente em termos de tempo e recursos, tornando-o ideal para grandes conjuntos de dados categóricos e para aplicação em tempo real.

Capacidade de Generalização: Embora limitado em dados complexos e com variáveis correlacionadas, o Naive Bayes é útil para problemas de classificação binária, como identificar a propensão ao uso do app.

# Aplicação dos Modelos Selecionados

Cada modelo foi treinado e avaliado de acordo com o processo a seguir para prever a probabilidade de adesão ao app Khiata e melhorar as recomendações de "matches":

Divisão dos Dados: O conjunto de dados foi dividido em 70% para treino e 30% para teste. Essa divisão fornece uma base ampla para treinar os modelos, mantendo uma amostra representativa para avaliar a eficácia e garantir que o modelo possa generalizar para novos dados.

Treinamento: Para cada modelo, aplicamos o GridSearchCV para otimizar os hiperparâmetros. Além disso, utilizamos o SMOTE (Synthetic Minority Over-sampling Technique) para balancear as classes e evitar que o modelo favoreça uma classe em detrimento da outra.

Métricas de Avaliação: Foram utilizadas as métricas Acurácia, Recall, Precisão e F1-score para avaliar o desempenho dos modelos:

* Acurácia mede a proporção de previsões corretas.
* Recall avalia a capacidade do modelo de identificar a classe "Sim" (usuários interessados), essencial para capturar o público-alvo do Khiata.
* Precisão verifica quantas das previsões positivas são realmente corretas.
* F1-score oferece uma média harmônica entre precisão e recall, proporcionando uma visão mais equilibrada da performance.

# Análise dos Resultados

Abaixo estão os resultados de desempenho e análise de cada modelo:

KNN (K-Nearest Neighbors):

Métricas de Avaliação: Obteve uma acurácia moderada, mas mostrou limitação em recall, especialmente em grandes volumes de dados.

Discussão dos Resultados: O KNN apresentou desempenho adequado como baseline para segmentar clientes com perfis similares. No entanto, devido ao custo computacional elevado ao lidar com grandes volumes, ele é mais adequado para conjuntos de dados menores.

Equilíbrio entre Precisão e Recall: Houve um certo desbalanceamento entre precisão e recall, com melhor desempenho para identificar clientes que não usariam o app ("Não").

Erro e Overfitting: KNN mostrou certa dificuldade de generalização, especialmente em dados de alta dimensionalidade.

Decision Tree:

Métricas de Avaliação: O modelo obteve um bom equilíbrio entre precisão e recall, com uma acurácia média de 0.65, proporcionando confiabilidade e boa capacidade de generalização.

Discussão dos Resultados: A Decision Tree com critério Gini ajustado demonstrou-se ideal para identificar fatores que influenciam o uso do Khiata, apresentando bom desempenho em ambas as classes. Esse modelo destaca-se pela facilidade de interpretação, auxiliando no entendimento dos principais interesses do usuário.

Equilíbrio entre Precisão e Recall: Conseguiu um bom balanceamento entre ambas as métricas.

Erro e Overfitting: O modelo mostrou baixo risco de overfitting devido ao ajuste dos hiperparâmetros, mantendo bom desempenho em novos dados.

Naive Bayes:

Métricas de Avaliação: Apresentou precisão aceitável para dados categóricos, mas com enviesamento em favor da classe majoritária.

Discussão dos Resultados: O Naive Bayes é simples e rápido, ideal para identificar potenciais usuários do app. Entretanto, a suposição de independência das variáveis comprometeu a acurácia em dados complexos.

Equilíbrio entre Precisão e Recall: Teve uma ligeira tendência a obter alta precisão, mas recall menor, indicando dificuldade em identificar adequadamente os clientes da classe minoritária.

Erro e Overfitting: Como o modelo é menos complexo, ele não apresentou problemas significativos de overfitting, mas teve dificuldade em lidar com correlações entre variáveis.

# Comparação Entre Modelos

## Comparação Direta:

Decision Tree apresentou o melhor desempenho geral, mostrando-se equilibrado entre precisão e recall, além de ser fácil de interpretar.

KNN teve um desempenho aceitável como baseline, útil para segmentação de perfis, mas limitado para grandes volumes de dados.

Naive Bayes foi rápido e eficiente para dados categóricos, mas teve limitação em dados correlacionados.

## Forças e Fraquezas:

Decision Tree: Equilibrado e interpretável, com bom desempenho em categorias complexas e uma estrutura que facilita a visualização das decisões.

KNN: Útil para segmentação de perfis semelhantes, porém apresenta dificuldades computacionais em grandes conjuntos de dados.

Naive Bayes: Simples e eficiente para dados categóricos, mas limitado para dados correlacionados e complexos.

# Ajustes Necessários

Para aprimorar os resultados, foram feitas modificações na base de dados e nos modelos:

Problemas de Desempenho: KNN apresentou baixo recall para classes menos frequentes, enquanto Naive Bayes favoreceu a classe majoritária.

Dados Desbalanceados: Foi aplicado SMOTE para balancear as classes, melhorando o desempenho do modelo em identificar usuários interessados.

# Pré-processamento:

Limpeza de Dados: Remoção de outliers e erros de entrada.

Transformação de Dados: Aplicação de normalização para manter variáveis na mesma escala.

Engenharia de Características: Criação de novas variáveis para capturar melhor as preferências de consumo.

# Ajustes nos Modelos:

Hiperparâmetros: Foram ajustados para cada modelo usando GridSearchCV.

Validação Cruzada: Aplicada para avaliar o desempenho dos modelos de forma robusta e evitar overfitting.

# Impacto das Modificações

Cada modelo foi aprimorado com as mudanças realizadas:

Decision Tree: Com ajustes nos hiperparâmetros, o modelo melhorou sua precisão e capacidade de generalização, apresentando resultados mais equilibrados entre as classes.

Naive Bayes: Com a normalização e balanceamento de dados, o modelo conseguiu melhorar o recall, embora continue sendo limitado para dados correlacionados.

KNN: Com ajustes nos parâmetros de vizinhança e balanceamento de classes, houve melhora na precisão, embora o custo computacional continue sendo um fator limitante.

# Conclusão

A análise mostrou que o Decision Tree é o modelo mais adequado para o Khiata devido ao seu equilíbrio entre precisão e recall e sua facilidade de interpretação. Ele é ideal para compreender os fatores que influenciam o interesse dos usuários e oferece uma segmentação confiável, essencial para recomendar "matches" entre clientes e costureiras. O Khiata poderá, assim, utilizar os insights gerados para fortalecer a personalização de recomendações, alinhando os interesses dos usuários às características das costureiras e promovendo uma experiência satisfatória na plataforma.

Recomenda-se que o Khiata continue monitorando o desempenho dos modelos em novos dados e explore melhorias na base de dados e nas técnicas de pré-processamento para garantir a precisão e a relevância das previsões

**Gráficos feitos para a IA escolhida**

**A red and white flag

Description automatically generated**

**A blue squares with numbers

Description automatically generated**

A blue and orange squares

Description automatically generated

A red green and blue rectangles

Description automatically generated

A purple and white color

Description automatically generated

# Integração da IA no aplicativo

Para fazer a integração da IA, nós enviamos o resultado do formulário realizado pela pessoa ao mongoDB, e através deste Banco de Dados nós pegamos a última inserção realizada, utilizamos o pré-processamento que estava serializado para preparar as informações e realizar a previsão com a IA, retornando ao banco o resultado previsto e mostrando para o usuário a análise da nossa inteligência artificial.

# RPA para o banco