Análise de modelos de Associação e Classificação

Integrantes:
Hugo Da Costa Gomes; RA: 1282117256
Matheus Menezes da Silva Ramos; RA: 1282118603
Mateus Sousa Schmidt; RA: 1282124086
Arley Novais do Nascimento; RA: 1282124086
Lucas Martins Santos; RA: 1282016408
Álvaro de Moura Maia; RA: 12822131248
Pedro Evaristo Dantas Zaranza; RA: 1282116206
Sammuel Augusto de Queiroz Lima Alves; RA: 1282226748
Camille Maria Pessoa Queiroz; RA: 12823131203

RESUMO

Dado uma amostra pequena, de cerca de 1500 linhas e uma análise inicial, foram analisadas diferentes maneiras de se tratar uma pequena base de dados com modelos de associação com Algoritmo Apriori e de classificação com algoritmo de árvore de decisão.

Ânima Educação 2024

ÍNDICE

1. INTRODUÇÃO

Foi utilizado dos dados fictícios de um supermercado chamado SuperMarketPlus, onde são divididos entre 1500 linhas de dados em forma de dados categóricos:

1. Faixa Etária: Categorias de faixas etárias ('18-25', '26-35', '36-45', '46-55', '56-65', '66-75', '>75').
2. Gênero: 'Masculino', 'Feminino'.
3. Frequência de Compras: Frequência com que o cliente faz compras no supermercado ('Diária', 'Semanal', 'Quinzenal', 'Mensal').
4. Valor Médio de Compra: Faixas de valor médio gasto por compra ('<50', '50-100', '100-150', '150-200', '>200').
5. Categoria de Produto Favorita: Categoria de produto que o cliente mais compra ('Alimentos', 'Bebidas', 'Limpeza', 'Higiene Pessoal', 'Outros').

Foi utilizado o Google Colab como plataforma para desenvolver os códigos e análise e então postado o arquivo que está disponível no Github.com.

2.METODOLOGIA E ANÁLISE

Primeiro foi considerado erros e inconsistências dentro do banco de dados, buscando por espaços em branco e erros de grafia que foram aplicados dentro do código para buscar por esses erros.

Depois foi considerado que a análise estava sendo feita para a empresa SuperMarketPlus e como os dados poderiam levar à insights a partir de uma análise simples dos dados, primeiro comparando as relações entre si.

Porém, boa parte dos dados se mostrou inconclusivo uma vez que suas diferenças eram pequenas, mas que já servem para ter uma visão geral da clientela do da empresa.

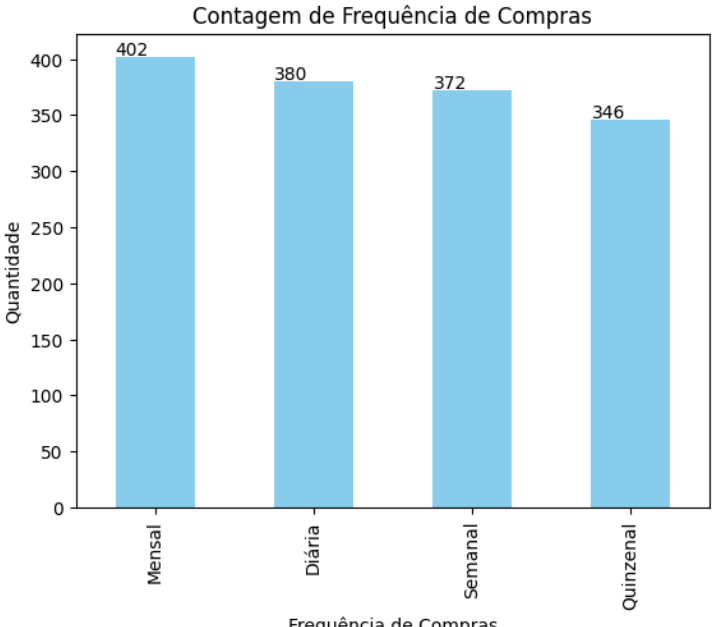


imagem 1

No gráfico da imagem 1 mostra como boa parte da clientela prefere fazer compras planejadas do que necessariamente compras diárias. Com os modelos será visto de maneira mais detalhada como os dados podem interagir entre si.

2.1 ALGORITMO APRIORI

O primeiro modelo a ser testado dentro dos dados foi o algoritmo apriori, que consiste em identificar padrões e regras dentro de um dado, o que pode ser usado para identificar mais facilmente padrões da clientela da empresa.

Primeiro foi feita uma conversão para uma lista de transações e transformação para o formato one-hot.

```
transactions = df.values.tolist()
#print para saber se funcionou corretamente
print(transactions[:5])
```

```
# Utilizando modelo Apriori

from mlxtend.frequent_patterns import apriori, association_rules

df_tr = pd.DataFrame(transactions)
df_onehot = pd.get_dummies(df_tr.stack()).groupby(level=0).sum()
```

E então, foi aplicado:

```
# Aplicar o algoritmo Apriori e gerar regras de associação
frequent_itemsets = apriori(df_onehot, min_support=0.1, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
print("_____")
print(rules)
```

As regras de associação geradas pelo algoritmo Apriori fornecem insights valiosos sobre o comportamento de compra dos clientes no supermercado, com foco em faixas de valor médio de compra, categorias de produtos e gênero dos clientes. As regras indicam que:

Pessoas do Gênero Masculino e Gastos Maiores: Homens tendem a gastar mais em compras, com uma presença significativa nas faixas de valor médio de compra entre 100-150, 150-200 e acima de 200. Isso sugere que estratégias de marketing focadas em produtos mais caros ou premium podem ser mais eficazes para o público masculino.

Gênero Feminino e Gastos Menores: Mulheres são mais propensas a fazer compras com valores menores, especificamente abaixo de 50. Além disso, elas têm uma presença significativa em compras

diárias e quinzenais. Isso pode indicar que ofertas e promoções de produtos essenciais e de baixo custo podem atrair mais clientes femininos.

Preferências por Categoria de Produtos:
Homens: Têm uma tendência maior a comprar alimentos, indicando que produtos alimentícios podem ser mais populares entre os clientes masculinos.
Mulheres: Preferem comprar bebidas, produtos de higiene pessoal e têm uma frequência de compras mais alta (diária e quinzenal). Isso sugere que campanhas publicitárias focadas em produtos dessas categorias podem ser mais eficazes para atrair clientes femininos.

Esses padrões podem ser utilizados para otimizar o estoque, direcionar campanhas de marketing específicas para diferentes segmentos de clientes e ajustar a oferta de produtos para atender melhor às preferências e comportamentos de compra de homens e mulheres.

Porém são resultados que devem ser analisados com cuidado e tempo, apesar de ter-se percebido alguns padrões nos dados, é recomendado grandes quantias e parâmetros aguçados quando trabalha com esses tipos de algoritmos, que como apresentados nos seus níveis acurácia do código.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(100-150)	(Masculino)	0.214000	0.505333	0.108667	0.507788	1.004858	0.000525	1.004987
1	(Masculino)	(100-150)	0.505333	0.214000	0.108667	0.215040	1.004858	0.000525	1.001324
2	(150-200)	(Masculino)	0.193333	0.505333	0.100667	0.520690	1.030388	0.002969	1.032038
3	(Masculino)	(150-200)	0.505333	0.193333	0.100667	0.199208	1.030388	0.002969	1.007337
4	(<50)	(Feminino)	0.199333	0.494667	0.107333	0.538462	1.088534	0.008730	1.094889
5	(Feminino)	(<50)	0.494667	0.199333	0.107333	0.216981	1.088534	0.008730	1.022538
6	(Masculino)	(>200)	0.505333	0.207333	0.108000	0.213720	1.030805	0.003228	1.008123
7	(>200)	(Masculino)	0.207333	0.505333	0.108000	0.520900	1.030805	0.003228	1.032492
8	(Alimentos)	(Masculino)	0.193333	0.505333	0.105333	0.544828	1.078155	0.007636	1.086768
9	(Masculino)	(Alimentos)	0.505333	0.193333	0.105333	0.208443	1.078155	0.007636	1.019089
10	(Bebidas)	(Feminino)	0.201333	0.494667	0.104667	0.519868	1.050945	0.005074	1.052487
11	(Feminino)	(Bebidas)	0.494667	0.201333	0.104667	0.211590	1.050945	0.005074	1.013010
12	(Diária)	(Feminino)	0.253333	0.494667	0.125333	0.494737	1.000142	0.000018	1.000139
13	(Feminino)	(Diária)	0.494667	0.253333	0.125333	0.253369	1.000142	0.000018	1.000048
14	(Higiene Pessoal)	(Feminino)	0.202000	0.494667	0.103333	0.511551	1.034133	0.003411	1.034568
15	(Feminino)	(Higiene Pessoal)	0.494667	0.202000	0.103333	0.208895	1.034133	0.003411	1.008716
16	(Outros)	(Feminino)	0.202000	0.494667	0.102667	0.508251	1.027461	0.002744	1.027624
17	(Feminino)	(Outros)	0.494667	0.202000	0.102667	0.207547	1.027461	0.002744	1.007000
18	(Quinzenal)	(Feminino)	0.230667	0.494667	0.116000	0.502890	1.016624	0.001897	1.016543
19	(Feminino)	(Quinzenal)	0.494667	0.230667	0.116000	0.234501	1.016624	0.001897	1.005009
20	(Masculino)	(Limpeza)	0.505333	0.201333	0.105333	0.208443	1.035314	0.003593	1.008982
21	(Limpeza)	(Masculino)	0.201333	0.505333	0.105333	0.523179	1.035314	0.003593	1.037426
22	(Masculino)	(Mensal)	0.505333	0.268000	0.136000	0.269129	1.004214	0.000571	1.001545
23	(Mensal)	(Masculino)	0.268000	0.505333	0.136000	0.507463	1.004214	0.000571	1.004323
24	(Semanal)	(Masculino)	0.248000	0.505333	0.126667	0.510753	1.010724	0.001344	1.011077
25	(Masculino)	(Semanal)	0.505333	0.248000	0.126667	0.250660	1.010724	0.001344	1.003549

Imagem 2

2.2 ALGORITMO DE ÁRVORE DE DECISÃO

Em seguida, foi feito uma análise com árvore de decisão, para tentar obter melhor precisão, foram utilizados 2 métodos, GridSearchCV e Optuna. Feito a decisão, foi decidido que iria usa os algoritmos para descobrir Categoria de Produto Favorita.

Primeiro houve a preparação dos dados e codificação das variáveis categóricas

```
X = df[['Faixa Etária', 'Gênero', 'Frequência de Compras', 'Valor Médio de Compra']]
y = df['Categoria de Produto Favorita']

# Codificar as variáveis categóricas usando LabelEncoder
label_encoders = {}
for column in X.columns:
    le = LabelEncoder()
    X[column] = le.fit_transform(X[column])
    label_encoders[column] = le

# Codificar a variável alvo
le_target = LabelEncoder()
y = le_target.fit_transform(y)
```

Então foi dividido o conjunto de dados entre treinamento e teste e foi definido o modelo de árvore e seus hiperparâmetros.

```
# Dividir o conjunto de dados em subconjuntos de treinamento e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Definir o modelo de árvore de decisão
dt = DecisionTreeClassifier(random_state=42)

# Ajustar os hiperparâmetros para otimizar a performance do modelo
param_grid = {
    'max_depth': [None, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100],
    'min_samples_split': [2, 5, 10, 50],
    'min_samples_leaf': [1, 2, 4, 7]
}
```

Utilizando do GridSearchCV para encontrar o melhor modelo

```
# Usar GridSearchCV para encontrar os melhores hiperparâmetros
grid_search = GridSearchCV(estimator=dt, param_grid=param_grid, cv=10, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Melhor modelo encontrado
best_dt = grid_search.best_estimator_

# Fazer previsões no conjunto de teste
y_pred = best_dt.predict(X_test)

# Calcular métricas de performance
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
```

Previsões e Métricas de Performance:
São feitas previsões (y_pred) com o modelo otimizado usando os dados de teste (X_test). Calculam-se métricas de performance como accuracy, precision, recall e f1-score utilizando as funções do sklearn.metrics. Um relatório de classificação é impresso, mostrando precision, recall, f1-score e suporte para cada classe de produto favorita.

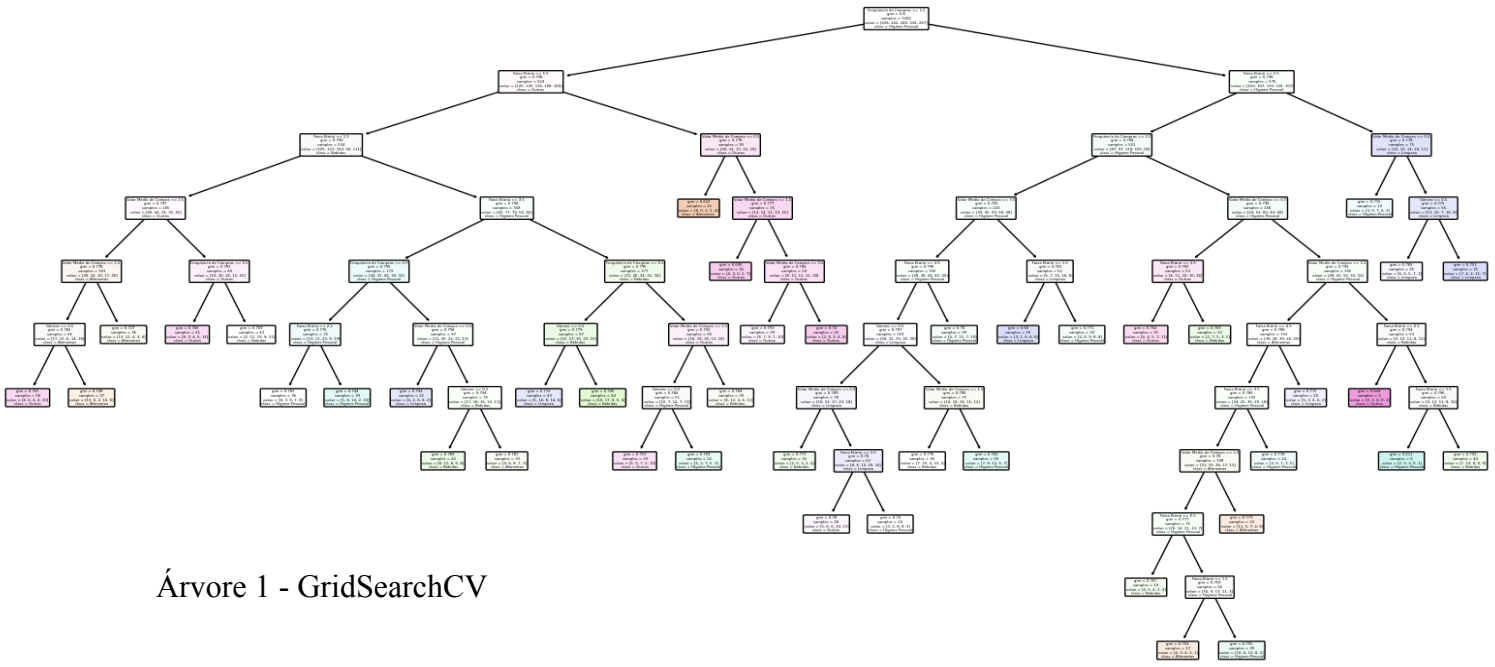
E então no final a árvore é plotada.

```
print(f"Accuracy: {accuracy}")
print(f"Precision: {precision}")
print(f"Recall: {recall}")
print(f"F1-Score: {f1}")

# Exibir o relatório de classificação
print(classification_report(y_test, y_pred, target_names=le_target.classes_))

# Visualizar a árvore de decisão
plt.figure(figsize=(20, 10))
plot_tree(best_dt, filled=True, feature_names=X.columns, class_names=le_target.classes_, rounded=True)
plt.show()
```

	precision	recall	f1-score	support
Alimentos	0.17	0.15	0.16	62
Bebidas	0.22	0.25	0.24	60
Higiene Pessoal	0.21	0.35	0.27	54
Limpeza	0.34	0.16	0.22	68
Outros	0.22	0.23	0.23	56
accuracy			0.22	300
macro avg	0.23	0.23	0.22	300
weighted avg	0.24	0.22	0.22	300



Árvore 1 - GridSearchCV

Os resultados indicam que o modelo de árvore de decisão construído não obteve uma performance elevada na classificação das categorias de produtos favoritas com base nas características dos clientes. A baixa acurácia e os valores modestos de precisão, recall e f1-score sugerem que as características escolhidas (Faixa Etária, Gênero, Frequência de Compras e Valor Médio de Compra) podem não ser suficientes para prever de forma precisa a categoria de produto favorita.

Com o método do Optuna, os resultados são melhores, mas não por muito:

	precision	recall	f1-score	support
Alimentos	0.24	0.16	0.19	62
Bebidas	0.24	0.37	0.29	60
Higiene Pessoal	0.20	0.24	0.22	54
Limpeza	0.33	0.31	0.32	68
Outros	0.32	0.21	0.26	56
accuracy			0.26	300
macro avg	0.27	0.26	0.26	300
weighted avg	0.27	0.26	0.26	300

3.CONCLUSÃO

Este estudo explorou o algoritmo Apriori para análise de regras de associação e árvores de decisão para classificação de categorias de produtos em dados transacionais de um supermercado. O Apriori revelou padrões de compra significativos entre diferentes grupos demográficos, destacando preferências específicas por categorias de produtos. Por meio da otimização dos hiperparâmetros, as árvores de decisão podem ser eficazes na previsão precisa da categoria de produto favorita dos clientes, mas poderia ficar melhor de fato com mais dados.

4.REFERÊNCIAS

Repositório Github: https://github.com/Coolffee/AD-BigD_A3UnP