Case Study

Discounts for close to expiration products (pink labels)

Alexandre Alves
Stefane Zandonadi
Andreia Campos
Sara Gomes
Filipa Carolino

Temas a abordar

1. Enquadramento
2. Preparação dos dados | Resumo
3. Dashboard
4. Identificação das melhores variáveis preditivas
5. Escolha e aplicação do modelo
6. Melhorias ao processo

Anexo I - Detalhe da preparação dos dados

Anexo II - Detalhe da identificação das melhores variáveis preditivas

1. Enquadramento

Problema: Prever se um produto com um *pink label atribuída* será ou não vendido (problema de classificação)

Objectivo: Evitar o desperdício alimentar e otimizar os lucros em produtos perecíveis

Dados disponíveis:

- 2 datasets iniciais (listagem dos produtos nos quais foi colocada a pink label – 150 054 linhas e 18 colunas - e listagem das lojas e das suas características);
- Considerámos uma junção das duas tabelas através da coluna idstore.

2. Preparação dos dados | Resumo

Melhorias na recolha de informação:

- Uniformização das marcas
- Uniformização das datas
- Coluna individual para o discount
- Uniformização das variáveis numéricas

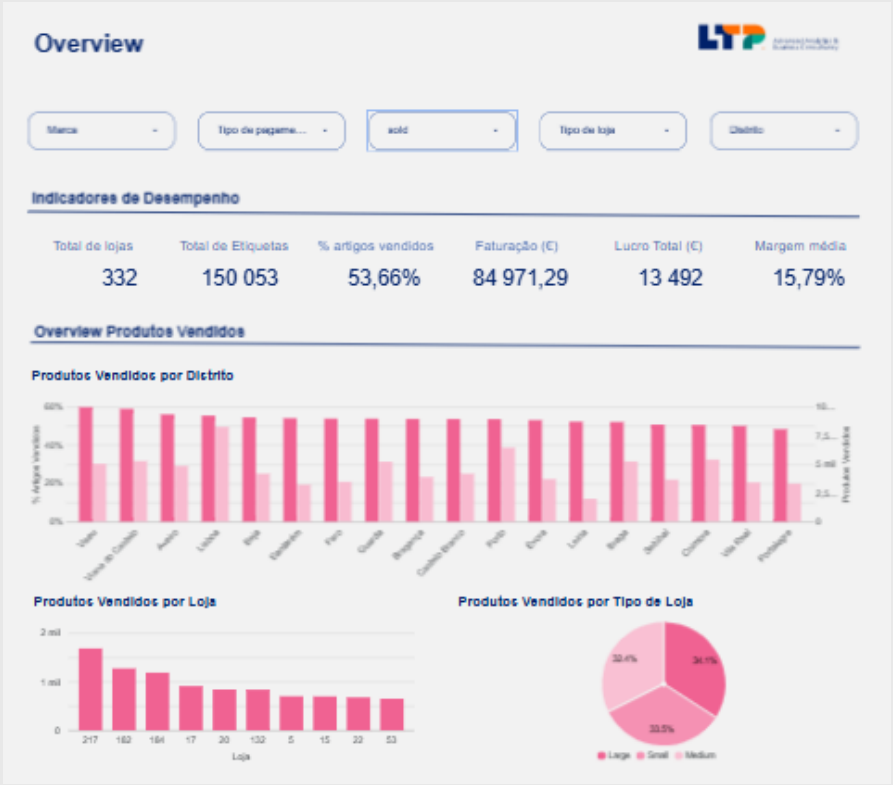
Variáveis relevantes/obrigatórias com valores em falta:

- Valor inicial do produto
- Valor do produto após desconto
- Valor do desconto
- Data da venda do produto

Inconsistências encontradas:

- Data de venda anterior à data de colocação da pink label
- Valores incorrectos na proporção de vida útil restante aquando da aplicação da pink label.

3. Dashboard



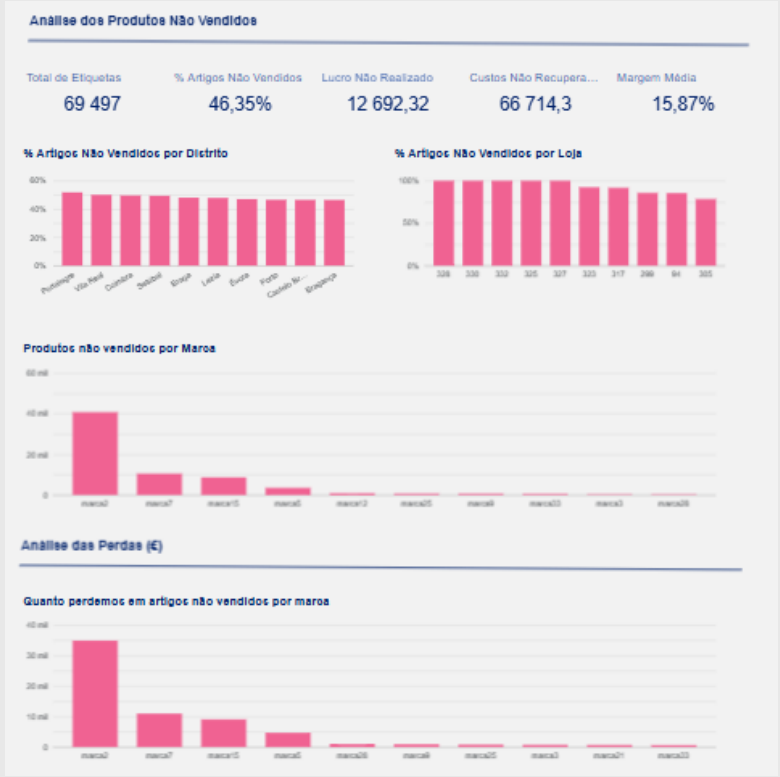


Imagem ilustrativa

4. Identificação das melhores variáveis preditivas

Variáveis categóricas:

Variáveis	Chi-Square	Incluir no modelo?
brand	5 Sim	Não – variável desbalanceada
district	5 Sim	Sim
labelling_day_of_week	5 Sim	Sim
labelling_day_8	5 Sim	Sim
labelling day 15	5 Sim	Sim
labelling day 23	5 Sim	Sim
type	0 Sim	Não

- **Brand** não foi considerada - variável desbalanceada
- **Selling_day_of_week** não foram considerados - geravam data leakage

4. Identificação das melhores variáveis preditivas

Variáveis numéricas:

Variável	RFE	Lasso	DT	Incluir no modelo?
Selling_square_ft	5 Sim	5 Sim	5 Sim	Sim
labelling_day	5 Sim	5 Sim	5 Sim	Tentar com e sem
weight (g)	5 Sim	5 Sim	5 Sim	Sim
Margem_num	0 Sim	5 Sim	5 Sim	Tentar com e sem
perc_expiring_sku	5 Sim	5 Sim	5 Sim	Sim
discount	0 Sim	0 Sim	0 Sim	Sim, tendo em conta o conteudo do problema
new_pvp	5 Sim	5 Sim	5 Sim	Sim

- A variável **discount** foi considerado uma variável com baixa relevância contudo considerámos uma variável relevante tendo em conta o conteúdo do problema

4. Identificação das melhores variáveis preditivas

CONCLUSÕES:

Variáveis a incluir no modelo	Descrição
selling_square_ft	Área de venda disponível
new_pvp	Preço identificado na etiqueta rosa
weight (g)	Peso de cada SKU
perc_expiring_sku	Proporção da validade restante do item quando a etiqueta foi impressa
Margem_num	Valor numérico da margem bruta
discount	Valor do desconto aplicado
district	Localização do distrito da loja
labelling_day_of_week	Dia da semana em que a etiqueta foi emitida.
labelling_day_8	Etiquetas emitidas até o dia 8 do mês (inclusive).
labelling_day_15	Etiquetas emitidas entre os dias 9 e 15 do mês.
labelling_day_23	Etiquetas emitidas entre os dias 16 e 23 do mês.
sold	Variável preditiva. (=1) se a etiqueta foi vendida antes da data de validade, (=0) caso contrário

5. Escolha e aplicação do modelo

Divisão dos dados:

- Treino (90%)
- Teste (10%)

Modelos utilizados:

- Logistic Regression (LR)
- K-Nearest Neighbors (KNN)
- Decision Tree(DT)

Resultados (selecção do modelo | pré otimização)

	Train	Validation
LR	0.667+/-0.0	0.667+/-0.0
KNN	0.799+/-0.0	0.727+/-0.0
DT	0.666+/-0.02	0.666+/-0.01

Otimização:

- O KNN é um algoritmo que **classifica um novo ponto de dados com base nos 'k' pontos mais próximos (vizinhos)** no conjunto de treino.
- Neste caso experimentamos diferentes números de vizinhos e escolhemos o melhor compromisso entre os resultados em treino e os resultados em validação.

Resultados (selecção do modelo | pós otimização)

	Train	Validation	Teste
Best LR	0.698+/-0.0	0.698+/-0.0	-
Best KNN	0.803+/-0.0	0.738+/-0.0	0.725224097

5. Melhorias ao processo

Informação adicional:

- Nomes das marcas e dos sku de forma a permitir uma eventual categorização
- Identificação da data de referência da extracção da informação
- Informação de quando é que o produto foi colocado na prateleira

Outras melhorias:

- Maior capacidade computacional para correr modelos mais exaustivos e com maior rapidez

Anexo I

Detalhe da preparação dos dados

Preparação dos Dados | Formatação

Erros encontrados	Alteração	Soluções utilizadas
Verificação de duplicados (33 740)	Não	Não eliminámos os duplicados porque considerámos que podiam ser de facto operações distintas
Formatação “Brand”	Sim	Uniformização das categorias com o mesmo significado (ex: “marca 2” e marca2”)
Formatação “old_pvp”	Sim	Substituir “,” por “.”. Arredondar para 2 casas decimais. Alterar o tipo de variável.
Formatação “new_pvp”	Sim	Separar o new_pvp do discount (criar uma coluna adicional “discount”). Substituir “,” por “.”. Alterar o tipo de variável.
Nova coluna “discount”	Sim	Uniformização dos valores (ex: 20% e 0.2)
Formatação “profit”	Sim	Alterar o tipo de variável. Arredondar para 2 casas decimais.
Formatação da “margem_num”	Sim	Transformação dos valores em número (0.2) em vez de percentage (20%).
“Perc_expiring_sku” > 1	Sim	Todas as observações dos skus 122 e 134 com este erro, e alterámos pela média dos perc_expiring_sku.
“expiring_date”, “labeling date”, “selling_date”	Sim	Uniformização das datas ao nível dos separadores (ex: 12/5/2025 e 12-5-2025) e ao nível da orientação da data (ex: dd/mm/aaaa e mm/dd/aaaa). Alterar o formato das variáveis.
“sold”	Sim	Alterámos a variável sold para boliano.
“weight”	Sim	Alterar o tipo da variável para float.
“idstore “ e “sku”	Sim	Alterar o tipo da variável para object.

Preparação dos Dados | Novas colunas

Novas colunas	Descrição
“selling_day”	Dia do mês (ex: 1,2,3) da selling date.
“Selling_day of the week”	Dia da semana (ex: Monday, Tuesday) da selling date.
“labelling_day”	Dia do mês (ex: 1,2,3) da labeling date.
“Labeling_day of the week”	Dia da semana (ex: Monday, Tuesday) da labeling date.
“vida_util apos label”	Expiring date – Labeling date.
“dias_sell_labeling”	Selling date - Labelling date
'labelling_day_8’	['labelling_day'] <= 8
'labelling_day_15’	['labelling_day'] <= 15 & ['labelling_day'] > 8
'labelling_day_23’	['labelling_day'] <= 23 & ['labelling_day'] > 15

Preparação dos Dados | Outliers

Erros encontrados	Alteração	Soluções utilizadas
“old_pvp” > 400 (2 produtos em específico – sku 4 e sku 108)	Sim	Sku 4 substituímos pela moda do old_pvp desse sku. Sku 108 tinha apenas uma observação e decidimos eliminar essa observação.
“new_pvp” um valor de 250	Sim	Correspondia à linha com o olp_pvp superior a 400. Desta forma substituímos tb pela moda.
“profit” 13 observações com profit sup 0.8 (sku 201 e 33)	Não	Fizemos uma análise detalhada das observações e considerámos que estes valores podiam ser reais, desta forma decidimos não os substituir.
“discount” com valor 0.99 (sku 4 e marca 3)	Sim	Considerando as várias possibilidades de discount destes sku’s, considerámos a moda do discount destes sku’s.
1491 observações com “weight” > 400	Não	Apesar de ser um valor significativamente superior aos restantes, verificámos que todas as observações estavam associadas ao mesmo sku (produto), desta forma decidimos manter.

Preparação dos Dados | Missing Values

Erros encontrados	Alteração	Soluções utilizadas
“payment_method “	Sim	Substituímos os valores nulos por “not sell”
“selling_day”, “Selling_day of the week”, Selling date	Não	Produtos não vendidos
“per_expiring_sku” tinha nulos para Sku 176, 149 e 45	Sim	Substituímos pela mediana do per_expring_sku.
“weight” 428 nulos	Sim	Substituímos utilizando o KNNimputer
“old_pvp” 22 nulos	Sim	Substituímos pelo New_pvp/(1-discount)
“discount” ou “new_pvp” nulo	Sim	Cálculámos o new_pvp=profit/margin e discount = 1- (new_pvp/old_pvp).
“selling_square_fit” 10 779 nulos	Sim	Substituímos pela média do tipo de loja por Distrito.

Preparação dos Dados | Validação das incoerências

Incoerências	Falhas	
Se Selling date preenchido e “Sold”=False	Não	Não foram encontrados erros
Selling date vazio e “Sold”=True	Sim	Alterámos para “Sold” =False quando selling data vazio
Cálculo do old_pvp * (1 - desconto) = new_pvp	Não	Não foram encontrados erros
new_pvp*margem = profit	Não	Não foram encontrados erros
selling_date inferior à labelling_date	Sim	Encontrámos 6 inconsistência. Nestes casos substituimos a selling date pela labelling date.
data de expiring superior à labelling	Não	Não foram encontrados erros

Anexo II

Detalhe da identificação das melhores variáveis preditivas

Best predictive variables | Feature Selection Categóricas

Metodologia: Chi-Square (Filter Method) com stratified K-Fold

(forma eficaz de selecionar variáveis categóricas relevantes para a variável alvo binária, preservando o equilíbrio das classes em cada subdivisão da validação cruzada)

Variáveis consideradas: sold, brand, district, labelling_day_of_week, labelling_day_15, labelling_day_8, labelling_day_23, type

Objetivo:
- Ver quais variáveis categóricas são mais relevantes em todos os folds.
- Avaliar se uma variável é consistentemente importante, e não só por sorte num único split.
- Obter p-valores médios e filtrar por $p < 0.05$.

Resultados:

- Para todos os fold a variável “type” foi considerada irrelevante. Todas as outras foram consideradas relevantes.

Conclusões:

- Desconsideramos as variáveis: type, selling day of the week (decorrente do data leakage), brand (variável muito desbalanceada, grande concentração na marca 2);
- Manter: district, labelling_day_of_week, , labelling_day_15, labelling_day_8, labelling_day_23.

Best predictive variables | Feature Selection Numéricas

Metodologia: Variância

Variáveis consideradas: oldpvp, labelqty, weight (g), perc_expiring_sku, selling_square_ft, new_pvp, discount, Margem_num, vida_util_apos_label,labelling_day

Objetivo:

- Avaliar a relevância de variáveis numéricas com base na sua variabilidade
- Eliminar variáveis com variância muito baixa, que fornecem pouca ou nenhuma informação discriminativa.
- Reduzir dimensionalidade do conjunto de dados (menos variáveis = modelos mais simples e rápidos).
- Ajudar na seleção de features antes do treino de modelos (pré-processamento).
- Evitar "overfitting", removendo variáveis constantes ou quase constantes.
- Melhorar a qualidade dos dados, concentrando a análise nas variáveis mais informativas.

Resultados:

- A variável labelqty apresenta variância 0, o que significa que se mantém constante e não faz sentido incluir.

Best predictive variables | Feature Selection Numéricas

Metodologia: Spearman

Variáveis consideradas: oldpvp, weight (g), perc_expiring_sku, selling_square_ft, new_pvp, discount, Margem_num, vida_util_apos_label,labelling_day

Objectivo:

- Avaliar a associação monotônica entre variáveis numéricas e a variável target (mesmo que não seja linear).
- Identificar variáveis numericamente relevantes com base na força e direção da correlação.
- Selecionar variáveis explicativas úteis para modelos de machine learning, descartando as que não têm correlação com o target.
- Classificar variáveis por grau de associação, permitindo ranking de importância inicial.
- Funciona bem com variáveis ordinais ou transformadas, não exige distribuição normal.
- Complementa outros métodos de seleção de variáveis (como chi-square ou variância).

Resultados:

- Correlação forte entre o perc_expiring_sku e o vida_util_apos_label e entre o new_pvp e old_pvp, desta forma só faz sentido considerar uma variável de cada grupo.
- Neste caso deixámos de considerar a old_pvp e a vida_util_apos_label.

Best predictive variables | Feature Selection Numéricas

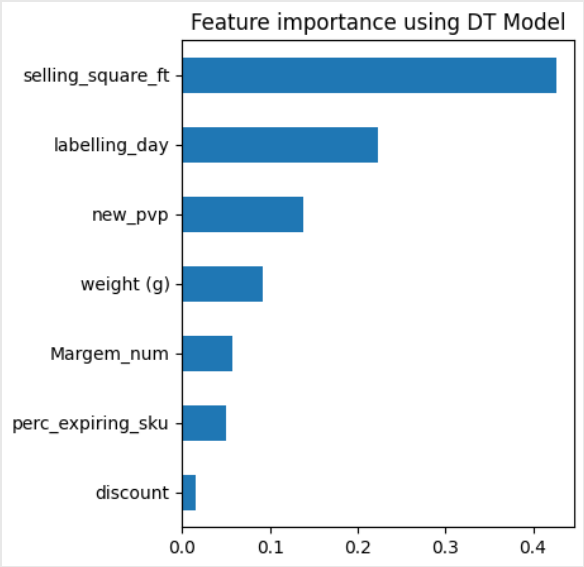
Metodologia: Árvore de Decisão, Recursive Feature Elimination (RFE) e Lasso

Variáveis consideradas: weight (g), perc_expiring_sku, selling_square_ft, new_pvp, discount, Margem_num, labelling_day

Objetivo:

- Classificar ou prever valores com base em regras simples (ex: "se... então...").
- Segmentar os dados automaticamente, criando grupos com características semelhantes.
- Dividir o espaço de decisão em regiões onde o target tem comportamento consistente.
- Aprender regras de decisão a partir dos dados de treino, sem necessidade de fórmulas estatísticas complexas.
- Modelar relações não lineares entre variáveis de forma natural.
- Servir como bloco base para modelos mais complexos como Random Forest ou Gradient Boosting.
- Fazer seleção implícita de variáveis → variáveis irrelevantes tendem a ser ignoradas pela árvore.
- Gerar explicações interpretáveis → cada decisão é visível como um caminho lógico (ótimo para relatórios e reguladores).

Resultados:



Conclusões:

- Apesar de este método desconsiderar as variáveis discount, decidimos manter visto que é uma variável relevante para o problema em questão

Best predictive variables | Feature Selection Numéricas

Metodologia: Recursive Feature Elimination (RFE)

Variáveis consideradas: weight (g), perc_expiring_sku, selling_square_ft, new_pvp, discount, Margem_num, labelling_day

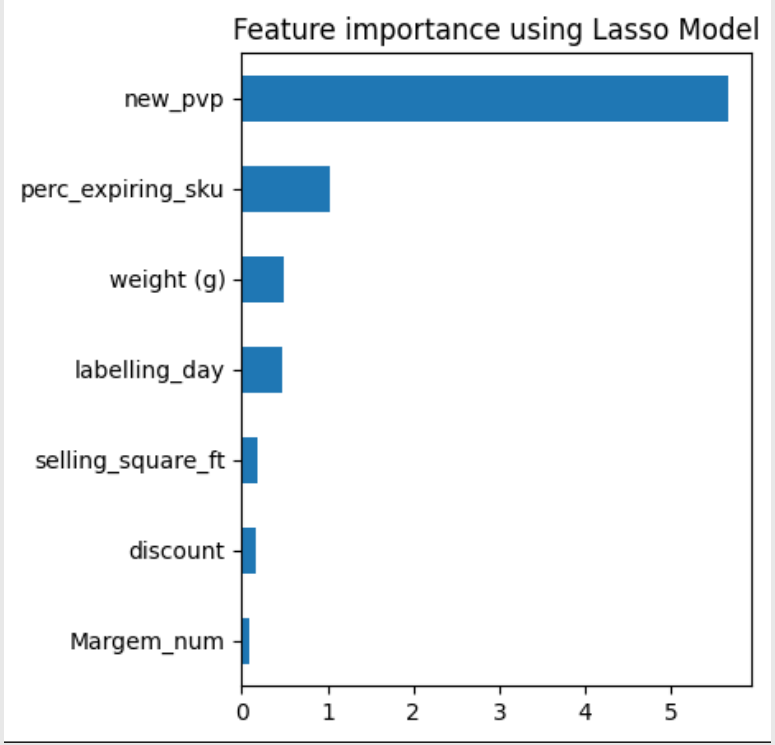
Objetivo:

- Selecionar as variáveis mais relevantes ao eliminar recursivamente as menos importantes com base no desempenho do modelo.
- Reduzir a complexidade do modelo, removendo variáveis redundantes ou irrelevantes.
- Melhorar a performance e a generalização do modelo, evitando overfitting.

Conclusões:

- Desconsiderar a variável margem_num

Resultados:



Best predictive variables | Feature Selection Numéricas

Metodologia: Lasso

Variáveis consideradas: Selling_square_ft, labelling_day, oldpvp, weight (g), Margem_num

perc_expiring_sku, labelqty

Objetivo:

- Selecionar variáveis automaticamente, forçando os coeficientes menos relevantes a zero (feature selection embutida).
- Reduzir o risco de overfitting, penalizando modelos muito complexos com muitos coeficientes.
- Melhorar a interpretabilidade do modelo, mantendo apenas as variáveis mais influentes.

Best predictive variables | Feature Selection Numéricas

Resultados:

Variável	Relevante?
Weight(g)	Sim
Perc_expiring_sku	Sim
Selling_square_ft	Sim
New_pvp	Sim
Discount	Não
Margem_num	Não
Labelling_day	Sim

Conclusões:

- A variavel margem_num foi considerada irrelevante. Apesar do discount também ter sido considerado irrelevante, decidimos mantê-lo, visto que é uma variável relevante para o problema em questão

Anexo III

Detalhe da Modelação e Otimização dos modelos

Modelação

Modelos Utilizados:

Logistic Regression: algoritmo utilizado para problemas de classificação. Consiste em combinação linear, semelhante à regressão linear mas sendo depois combinada com uma conversão para uma função sigmóide. No fundo, iremos querer ver quais são os melhores pesos a atribuir (ordenada na origem e declive da reta) e depois transformar numa curva em que o que estiver acima de determinado valor (geralmente 0.5) é classificado como sim e abaixo como não. Dá-nos a probabilidade de uma observação pertencer a uma certa classe.

KNN: K-Nearest Neighbors - funciona com a premissa de ver pontos de dados semelhantes que estão próximos uns dos outros sendo que temos de escolher depois quantos "vizinhos" queremos e de que forma queremos calcular a distância, escolhendo depois a classe a que uma observação pertence (em problemas de classificação) através da moda da classe dos vizinhos.

Decision Tree: Árvore de Decisão: divide os dados em subconjuntos com base nos valores dos atributos, criando uma estrutura em forma de árvore que pode ser usada para classificar ou prever valores para novas instâncias. Começa com um só nó com todos os dados e vai vendo qual o melhor atributo para dividir em novos ramos e vai por aí fora até obtermos x número de folhas através das quais classificaremos onde se enquadrarão as nossas novas observações.

Otimização

Otimização dos Modelos:

Logistic Regression: para este, utilizámos duas formas. O RandomizedSearchCV e o GridSearchCV. No fundo, fazem o mesmo, no entanto, demos prevalência aos resultados do Grid Search uma vez que este faz uma busca exaustiva (experimenta todas as combinações possíveis e vê qual nos dá um melhor resultado). Os 4 parâmetros que procurámos ajustar foram o *C* (força da regularização), a *penalty* (tipo de regularização), o *solver* (algoritmo de otimização a usar) e o *max-iter* (nº máximo de iterações que pretendemos).

Best Hyperparameters for Logistic Regression: {'solver': 'saga', 'penalty': 'l1', 'max_iter': 300, 'C': 0.001}

KNN: No nosso caso, passou essencialmente pela escolha do número de vizinhos (k) que queríamos. O melhor acabou por ser apenas 1 vizinho.

Decision Tree: Utilizámos também o RandomizedSearchCV e o GridSearchCV, dando uma vez mais prevalência aos valores do Grid Search.

Os 3 hiperparâmetros que procurámos otimizar foram o *criterion* (função para medir a qualidade de uma divisão, ou seja, ver qual é a variável que vamos usar para dividir a árvore), a *max_depth* (qual a profundidade máxima da árvore) e o *min_samples_split* (que é o nº mínimo de observações que queremos que esteja em cada folha final).

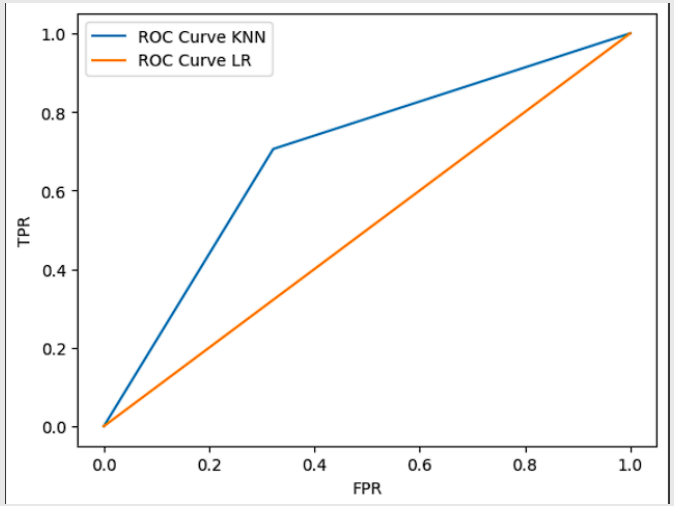
Best Hyperparameters: {'criterion': 'gini', 'max_depth': 3, 'min_samples_split': 2}

Escolha do melhor modelo

Escolha dos Modelos Otimizados:

No final, descartamos a Decision Tree uma vez que era a que nos estava a dar os piores resultados e voltamos a correr os modelos com os parâmetros obtidos na otimização. Para isso, pegamos nos 90% dos dados que colocamos para treino e dividimos em 80% para treino e 20% para validação.

Voltamos a rodar o código para os modelos do KNN e LR e obtivemos a confirmação de que o KNN seria o melhor. Fizemos também uma ROC Curve para visualizar também isto e confirmar (é o melhor modelo o que tiver uma maior área por baixo da curva):



Resultado

Aplicação do melhor modelo aos 10% de teste:

No final, pegamos então no nosso melhor modelo do KNN e aplicámos aos 10% que tínhamos deixado de parte no início para teste.

Neste caso não houve necessidade de alterar o threshold uma vez que só estamos a lidar com um vizinho. Isto faz com que sempre que for maior que 0, vá ser um Sim uma vez que a previsão de probabilidades contêm apenas 0's e 1's.

Obtivemos assim um F1-Score final de:

0.72522