Machine Learning Engineer Nanodegree
Capstone Project
Categorização de Produtos

Rafael Ferrari Zitto
07 de Abril de 2018

I. Definição

Visão Geral do Projeto

O comércio eletrônico é parte integrante da vida moderna, tanto para pessoas como para empresas. Tudo pode ser encontrado online, em grande variedade de marcas, modelos e preços.

Mas esta quantidade de opções traz um desafio: como organizar tudo isto? Tradicionalmente, lojas online utilizam uma estrutura hierárquica de categorias de produtos, como nos exemplos abaixo:

- *tv e home theater*
  - *tv*
    - *tv 4k*
      - ***Smart TV LED 58" Samsung 58mu6120 Ultra HD 4K***
- *Eletrodomésticos*
  - *geladeira / refrigerador*
    - *geladeira frost free*
      - ***Refrigerador Consul Frost Free Duplex 340 Litros***

Porém, apesar de parecer uma atividade trivial, classificar cada produto pode se tornar um desafio quando se trata de centenas de milhares deles, em centenas ou mesmo milhares de categorias[(2)].

Tudo fica ainda mais complexo quando se considera as atuais plataformas de marketplace, onde os mesmos produtos são cadastrados diversas vezes, por diferentes varejistas, muitas vezes de forma automatizada.

Apesar destas dificuldades, a categorização correta de produtos traz muitos benefícios, como:

- Navegação simplificada.
- Visualização e comparação de produtos similares em uma mesma página.
- Oferta de produtos e serviços relacionados (cross-selling/up-selling), tais como modelos mais sofisticados, garantia estendida e seguros.
- Geração de relatórios e dashboards gerenciais segmentados.

Com base neste cenário, este projeto se propõe a estudar estratégias de ***aprendizado de máquina supervisionado*** para a classificação automática de produtos em categorias, a partir de suas características conhecidas.

Descrição do Problema

Para atender a necessidade descrita, será realizada a **seleção**, **parametrização** e o **treinamento** de um algoritmo de classificação baseado em aprendizado de máquina.

Em execução, este algoritmo deve receber os dados de um produto e retornar a categoria do mesmo.

Figura 1 - Fluxo de execução

Dentre os dados de um produto, serão utilizados como entrada seu **nome**, sua **descrição** e o nome do seu **fabricante**.

A classificação resultante terá uma **estrutura hierárquica de categorias**, conforme padrão de mercado já exemplificado no tópico anterior.

A etapa de treinamento necessita de exemplos de produtos já classificados. Estes dados serão obtidos de um dataset disponibilizado publicamente pelo site Best Buy[6] (em inglês).

Métricas

Acurácia

Os modelos serão inicialmente avaliados por sua acurácia, por ter fácil interpretação e por constar em várias referências[1][2][3], o que possibilita a realização de benchmarks.

A acurácia calcula a fração de classificações corretas dentre todas as classificações realizadas. O valor da acurácia varia entre 0 e 1 (ótimo).

$$Acurácia = \frac{\Sigma\, Positivos\ verdadeiros + \Sigma\, Negativos\ verdadeiros}{\Sigma\, População\ total}$$

F1-score

Em conjuntos de dados desbalanceados, ou seja, onde as classes são observadas em quantidades desproporcionais, a métrica de Acurácia é prejudicada.

No caso em estudo é esperado que a quantidade de produtos em cada categoria (classe) varie significativamente. Por exemplo, podemos ter uma diversidade muito maior de produtos em "*Livros* > *Autoajuda*" do que em "*Smartphone* > *Iphone*".

Assim, será utilizada também a métrica F1-score, mais robusta para esta situação.

O F1-score é a média harmônica entre a precisão (precision) e a revocação (recall), o que penaliza tanto valores de falsos positivos, como valores de falsos negativos. O valor do F1-score também varia entre 0 e 1 (ótimo).

$$F1 = 2 \cdot \frac{Precisão \cdot Revocação}{Precisão + Revocação}$$

Onde:

$$Precisão = \frac{\Sigma\ Positivos\ Verdadeiros}{\Sigma\ Positivos\ Verdadeiros + \Sigma\ Falsos\ Positivos}$$

$$Revocação = \frac{\Sigma\ Positivos\ Verdadeiros}{\Sigma\ Positivos\ Verdadeiros + \Sigma\ Falsos\ Negativos}$$

Métricas Hierárquicas

Devido ao aspecto hierárquico da saída, as mesmas métricas também serão aplicadas nível a nível, permitindo mensurar como a Acurácia e o F1-score se deterioram com as progressivas segmentações dos produtos.

II. Análise

Exploraçãeo dos Dados

Os dados utilizados foram disponibilizados pela Best Buy, em uma base de dados relacional SQLite[6].

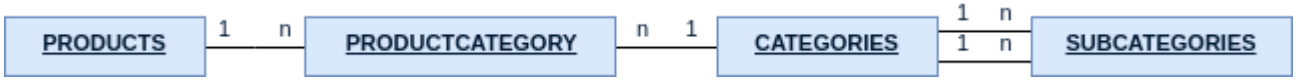

Figura 2 - Modelo de dados relacional da base SQLite

Por se tratar de uma base de dados relacional, os dados foram inicialmente desnormalizados e exportados para um arquivo CSV. O arquivo contém **42.747 linhas**. A *Tabela 1* exibe uma amostra dos dados.

name	description	manufacturer	cat_1	cat_2	cat_3	cat_4	cat_5	cat_6	cat_7	cat_8
KitchenAid - 36" Built-In Electric Cooktop - Black	Knob controls; 5 cooktop elements; 100-2700 …	KitchenAid	Cooktops	Ranges, Cooktops & Ovens	Appliances					
Duracell - AAA Batteries (4-Pack)	Compatible with select electronic devices; ...	Duracell	Alkaline Batteries	Household Batteries	Housewares	Connected Home & Housewares				
GoPro - Camera Mount Accessory Kit - Black	Compatible with most GoPro cameras; includes ...	GoPro	Handlebar /Seatpost Mounts	Action Camcorder Mounts	Action Camcorder Accessories	Camcorder Accessories	Cameras & Camcorders			

Tabela 1 - Amostra dos dados no arquivo CSV

Pode-se observar na amostra que os níveis de categoria de um produto estão ordenados do mais específico para o mais geral (ex.: *Cooktops* > … > *Appliances*). Esta ordem deve ser invertida durante o processamento dos dados.
Observa-se também a ocorrência de valores faltantes/nulos. O gráfico abaixo mostra este fato para o dataset completo, onde podemos destacar que:

- temos 1 produto sem nome e 1 produto sem fabricante
- todos os produtos possuem pelo menos 2 níveis de categoria
- menos de 1% dos produtos possuem 6 níveis de categoria
- nenhum produto possui 7 ou mais níveis de categoria

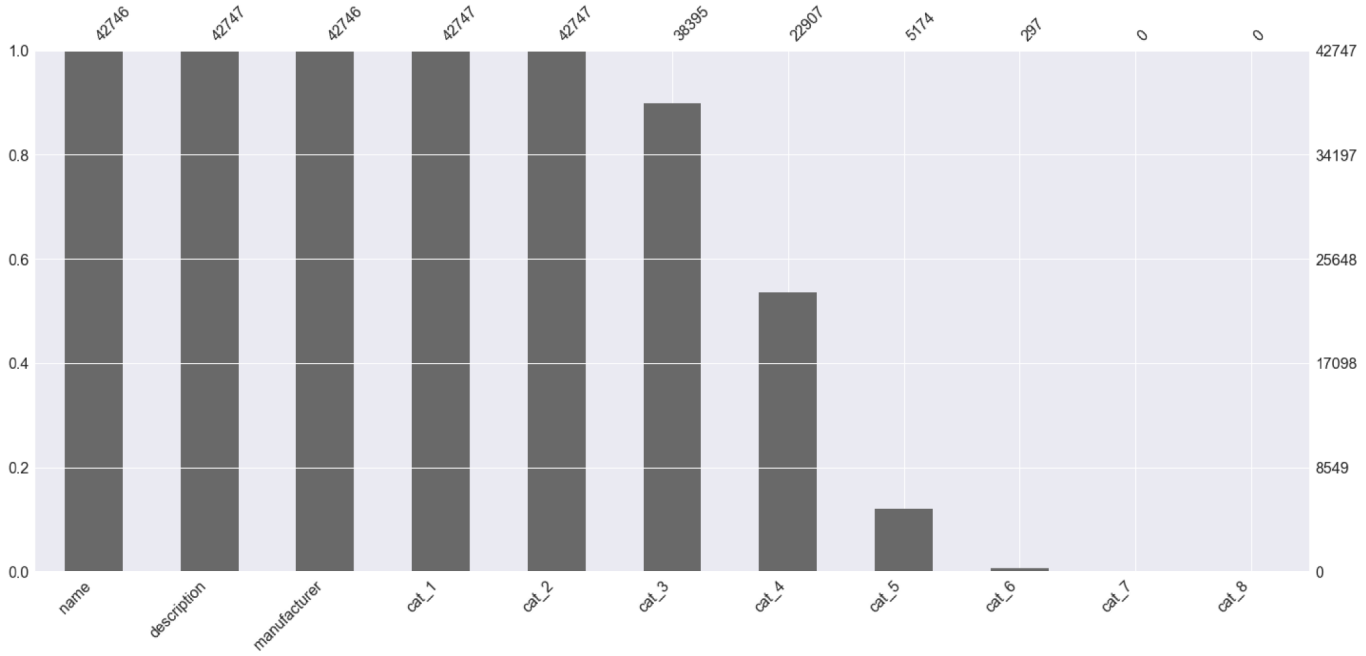

Figura 3 - Quantidade de valores não nulos por atributo

Com base nesta observação, os níveis 7 e 8 serão desconsiderados do estudo, pela total ausência de dados. Por simplicidade, o nível 6 também será desconsiderado. Com isto, a ***categoria completa*** de um produto será a **junção dos 5 níveis remanescentes**. Assim, no momento tem-se:

- Total de produtos: 42.747
- Total de categorias completas: 1.161

Investigando a quantidade de produtos contidos em cada categoria completa, foi possível verificar que a amostra é bastante desbalanceada. Encontra-se algumas com apenas 1 produto, assim como uma com mais de 3.000 produtos.

Média	**Desvio padrão**	**Mínimo**	**25%**	**50%**	**75%**	**Máximo**
37	138	1	3	11	34	3578

Tabela 2 - Produtos por categoria

Por se tratar tratar de um conjunto de dados disponibilizado como amostra pelo site Best Buy, representando uma pequena fração do seu catálogo completo, este desbalanceamento pode ser fruto do processo de amostragem utilizado em sua criação.

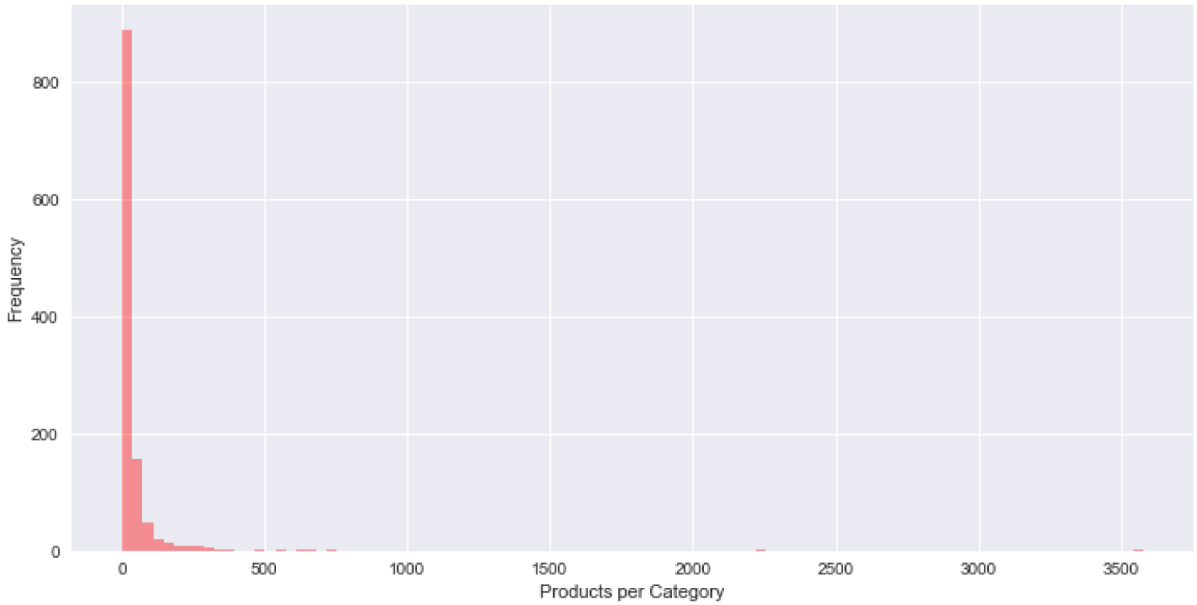

Figura 4 - Histograma - Qtd. Produtos por Categoria

Com base nisto, durante a etapa de pré-processamento foi decidido pela exclusão de categorias com menos de 50 produtos, assim como pela redução de categorias com mais de 500. Com isto altera drasticamente as análises subsequentes, todas já serão feitas com este novo subconjunto.

O conjunto final que será utilizado fica:

- **Total final de produtos: 25.115**
- **Total final de categorias: 197**

Média	**Desvio padrão**	**Mínimo**	**25%**	**50%**	**75%**	**Máximo**
127	105	50	62	84	152	500

Tabela 3 - Produtos por categoria após redução

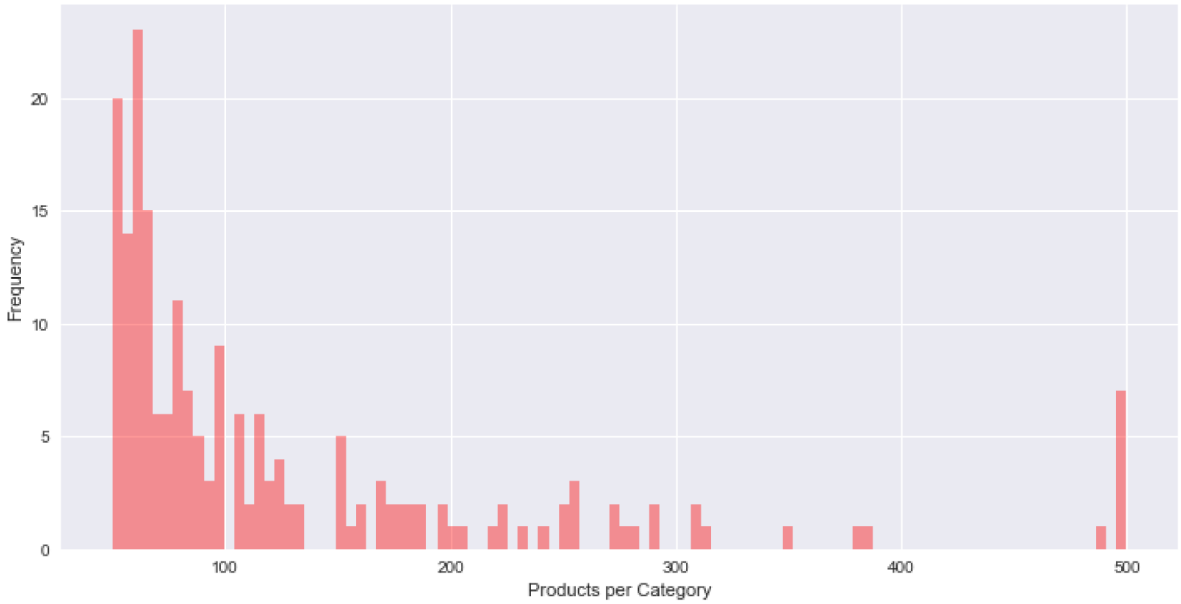

Figura 5 - Histograma - Qtd. Produtos por Categoria após a redução

Outra característica interessante é a quantidade de categorias *filhas* em que cada categoria *pai* se divide. Este valor será importante durante a classificação hierárquica discutida à frente, pois define o número de classes sendo considerado em cada nível de classificação. Na média este número é muito baixo.

Média	**Desvio padrão**	**Mínimo**	**25%**	**50%**	**75%**	**Máximo**
2.173653	2.091238	1	1	1	2	13

Tabela 4 - Subcategorias por categoria

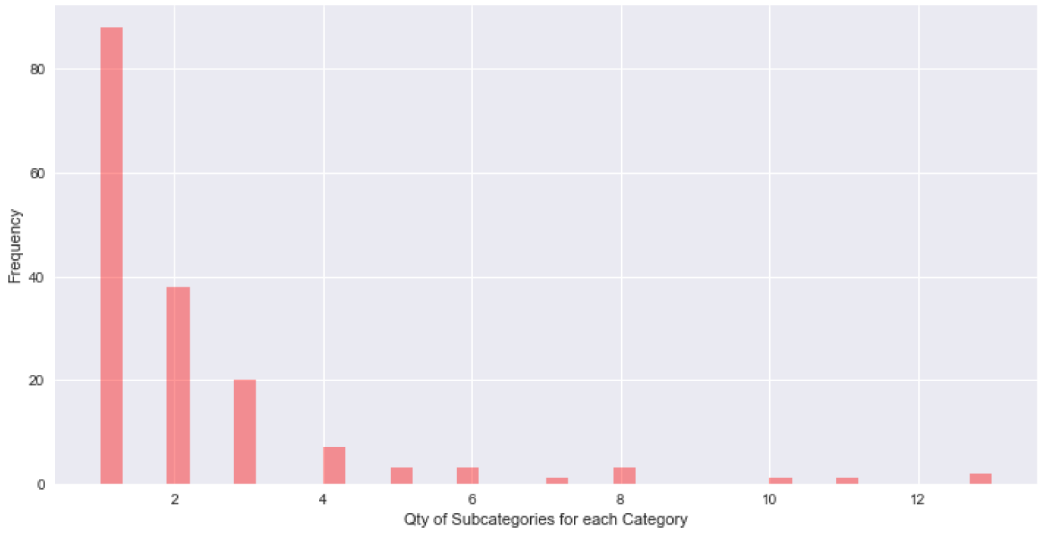

Figura 6 - Histograma - Quantidade de Subcategorias por Categoria

Finalmente, como os dados de entrada são compostos de texto, é interessante verificar o tamanho do vocabulário que cada atributo possui.

- Nome do produto: 11.045 palavras únicas
- Descrição do produto: 15.520 palavras únicas
- Nome do fabricante: 1.489 palavras únicas

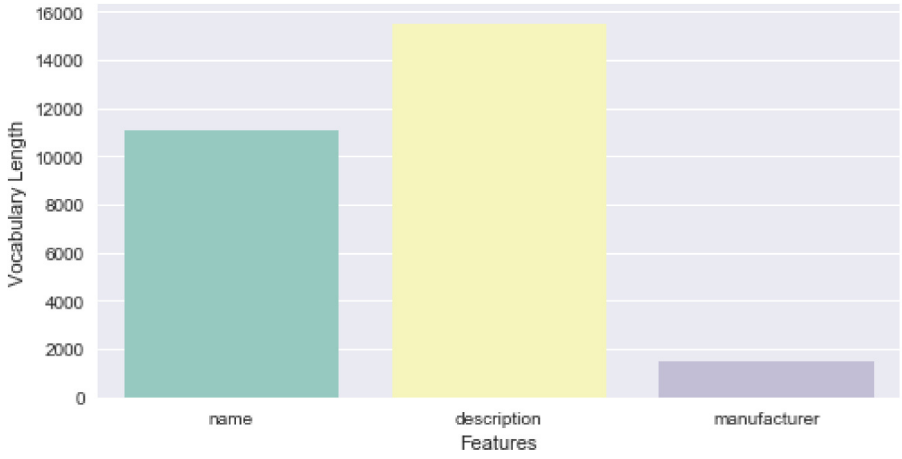

Figura 7 (ao lado) - Tamanho do Vocabulário por Atributo do Produto

Visualização Exploratória

Nesta visualização temos a representação hierárquica de todas as categorias, onde o círculo mais interno representa o nível 1, e o mais externo o nível 5. Cada nível é dividido por suas categorias, proporcionalmente a quantidade de produtos de cada. Por restrição de espaço, os nomes das categorias foram omitidos.

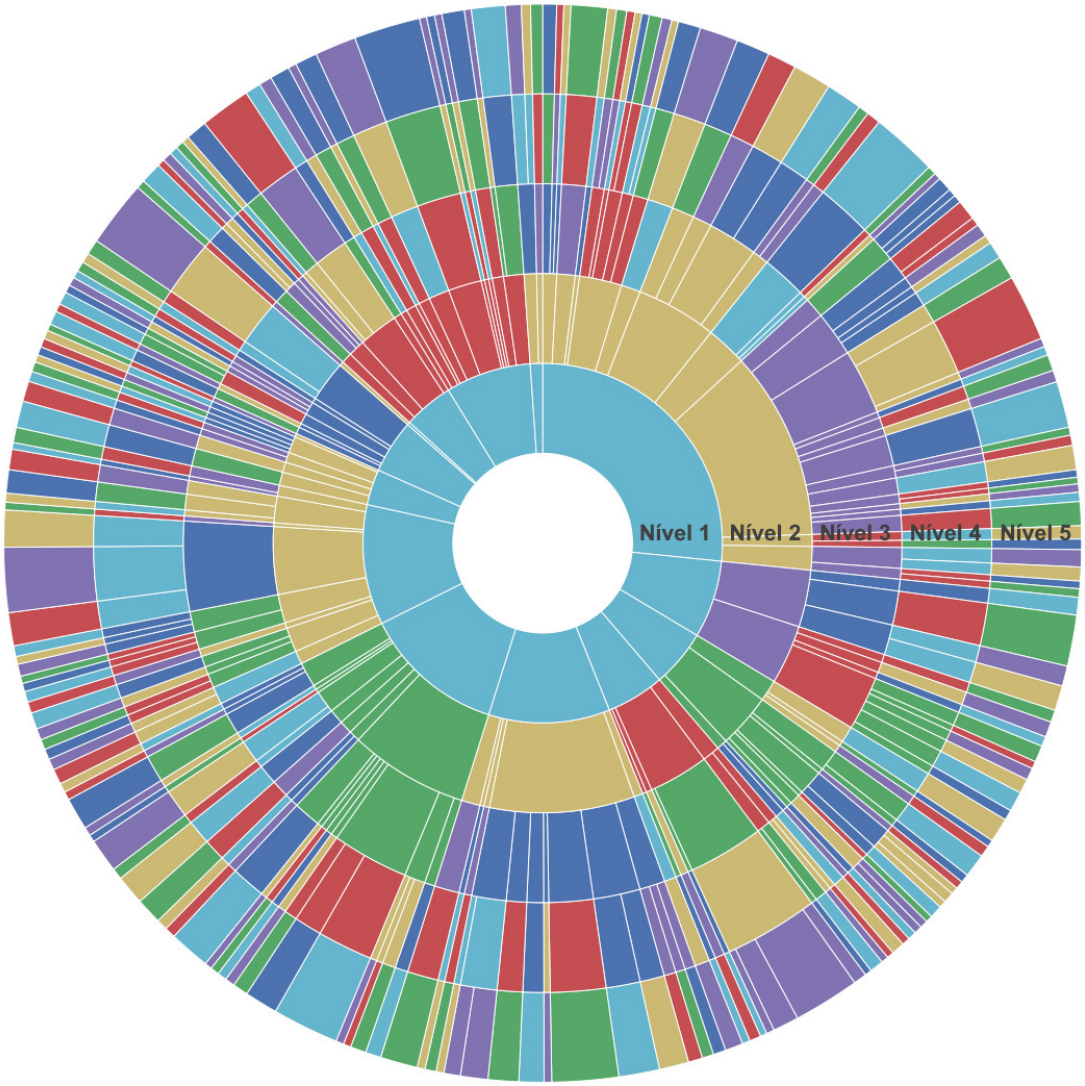

Figura 8 (ao lado) - Distribuição hierárquica das categorias

Nesta visualização é possível observar o desbalanceamento entre as categorias e a variação na quantidade de subcategorias por categoria.

A próxima figura exibe uma ampliação da anterior, sendo possível observar os nomes das categorias.

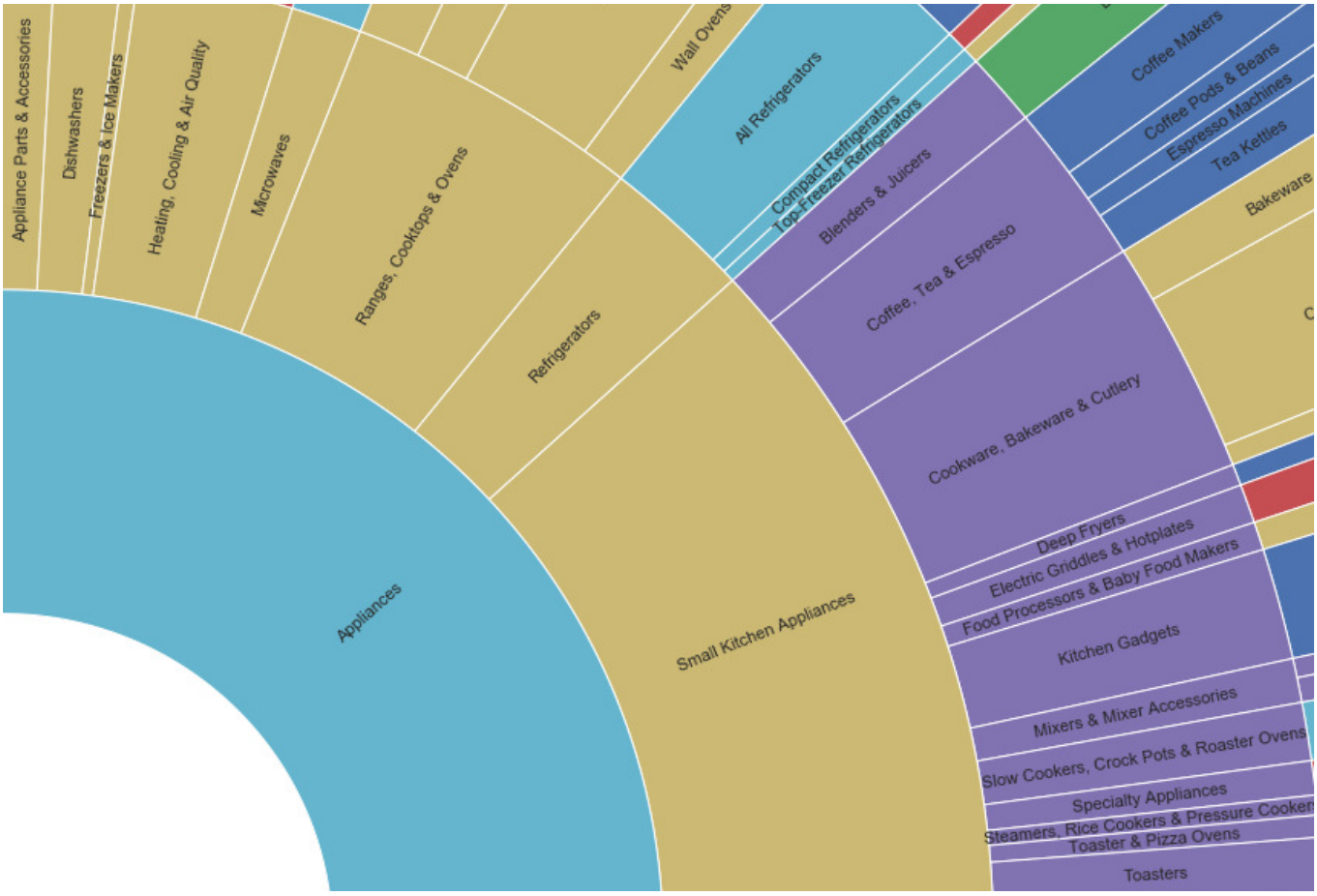

Figura 9 - Distribuição hierárquica das categorias - área ampliada

Algorítmos e Técnicas

Conforme já mencionado acima, o problema será tratado como classificação por aprendizagem supervisionada, utilizando-se um algoritmo indicado para este fim.

Alguns algoritmos foram pré-selecionados para avaliação. A pré-seleção considerou algoritmos que obtiveram bons resultados em outros estudos similares[1][2][3]. Segue breve justificativa para cada candidato:

Multinomial Naive Bayes

Simples e rápido, este algoritmo lida bem com grande quantidade de features, o que é o caso em estudo.
Por outro lado, pressupõe que todas as features são independentes, o que não é o caso para palavras em um texto.

Logistic Regression

Obteve a melhor performance em um dos estudos[2] e a probabilidade de cada classe é calculada naturalmente durante o processo de classificação.
É um modelo com “bias” alto, o que pode levar ao underfitting, ou seja, pode não capturar nuâncias relevantes no texto.

Decision Tree

Decision Trees já realizam implicitamente a seleção das features mais relevantes (“palavras”, no caso) e possui fácil interpretação.
Por outro lado, é um modelo suscetível ao overfitting, não generalizando bem.

Random Forest

Expande “Decision Tree”, criando uma composição (ensemble) de árvores treinadas com subconjuntos dos dados. Desta forma pode trazer melhores resultados, como, por exemplo, evitando o over-fitting.
Contudo, perde a interpretabilidade de uma Decision Tree.

SVC (linear kernel)

Lida bem com um grande volume de features e obteve a melhor performance em dois estudos[(1)(3)].
Como ponto negativo, é suscetível ao overfitting, e pode ter dificuldades em convergir no treinamento, dependendo do kernel utilizado.

A avaliação ocorrerá contra um subconjunto de dados, e cada algoritmo estará com seus parâmetros (hyperparameters) com valores *default*. O algoritmo com melhor F1-score será escolhido para ser utilizado no restante do estudo.

Como os dados de entrada são compostos de texto, será necessário a extração de features para que qualquer dos algoritmos acima seja utilizado. A extração será feita utilizando-se o método **TF-IDF** (term frequency – inverse document frequency).
Este problema apresenta um desafio (ou oportunidade) adicional, visto que o label tem uma estrutura hierárquica. Assim, algumas estratégias podem ser consideradas, conforme detalhado a seguir.

1. Classificador Global

Considera a categoria completa de um produto como sendo um label único, indivisível. Neste caso o conceito de hierarquia está sendo desconsiderado, e tem-se um treinamento único, com uma grande quantidade de classes de saída[(2)].

Figura 10 - Fluxo com classificador global

Relembrando a *Figura 8*, esta estratégia trabalha diretamente com as opções do círculo mais externo. A figura ao lado mostra a quantidade e proporção das classes (categorias) envolvidas na classificação de uma cafeteira (categoria em vermelho).

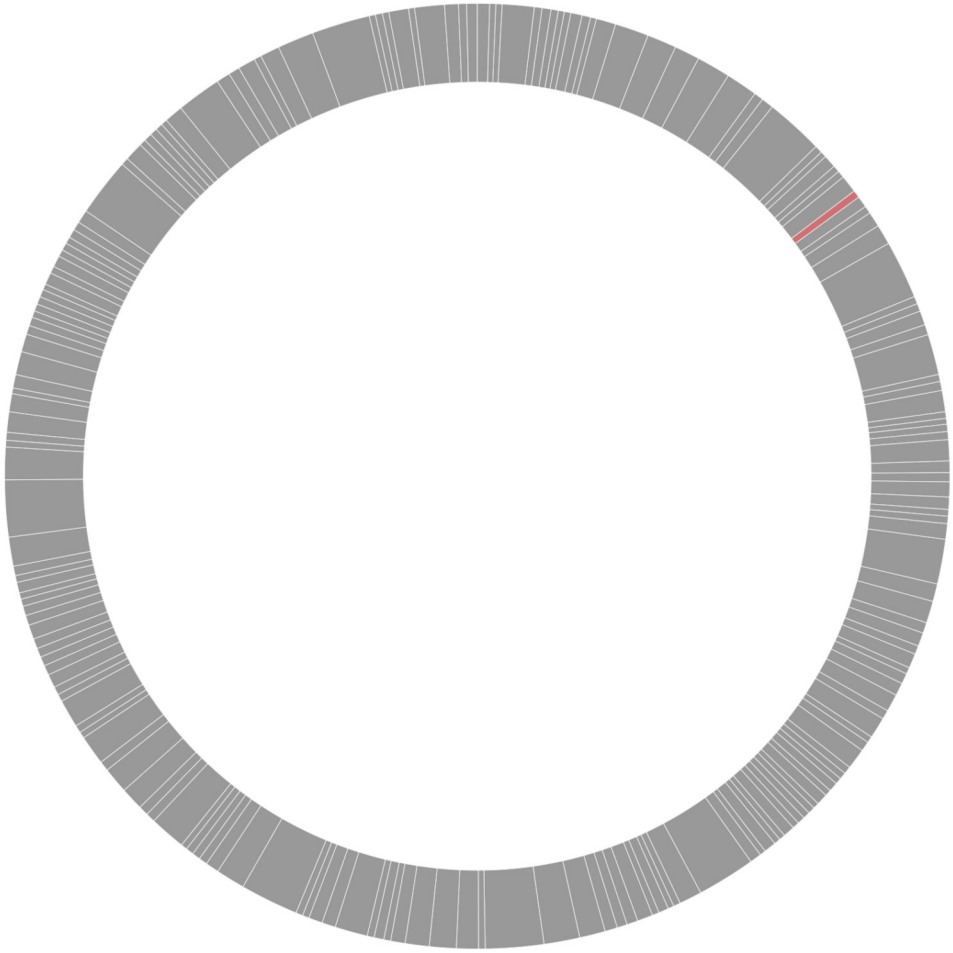

Figura 11 (ao lado) - Exemplo de classificação de uma cafeteira (classif. global)

2. Classificador Hierarquico

Considera cada nível da estrutura hierárquica individualmente. Neste caso teremos um treinamento para cada nó da hierarquia, e a predição ocorrerá de forma progressiva, nível por nível[3].

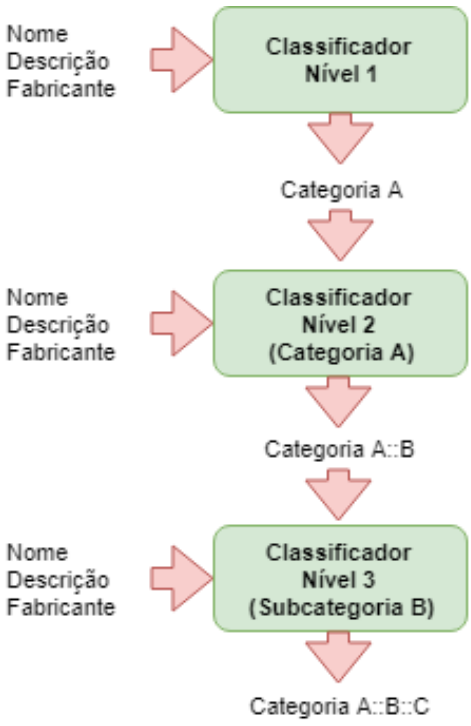

Figura 12 (ao lado) - Fluxo com classificador hierárquico

Relembrando novamente a *Figura 8*, esta estratégia trabalha círculo por círculo, iniciando do mais interno. As figuras abaixo mostram cada passo da classificação, onde fica evidente que a quantidade de possíveis classes é significativamente menor.

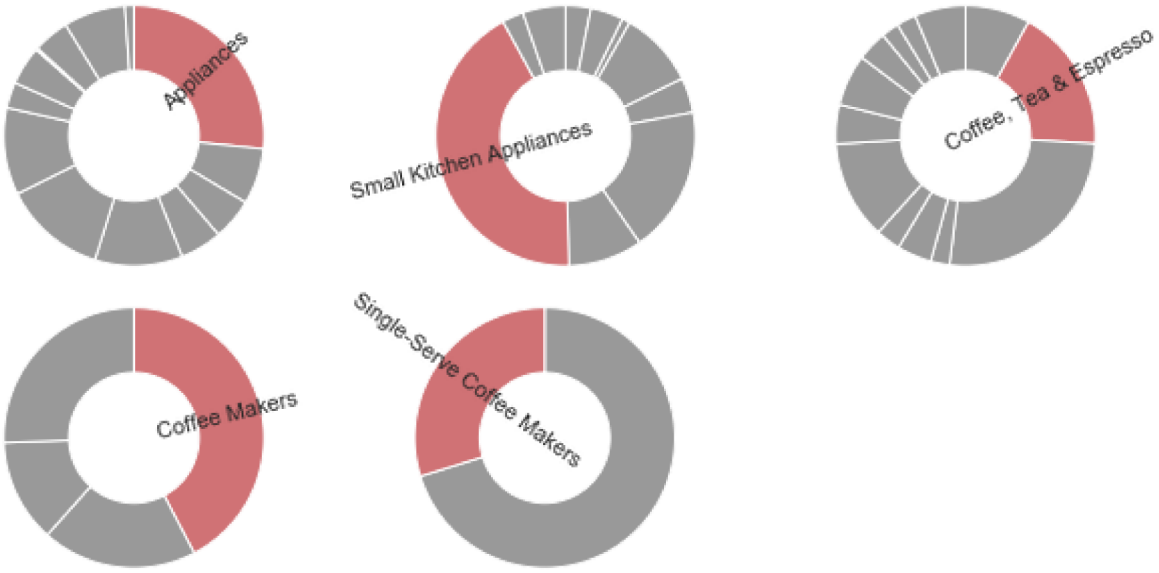

Figura 13 - Exemplo de classificação de uma cafeteira (nível por nível)

3. Classificador Hierarquico com Busca

A última opção é uma customização da anterior. Na anterior o classificador retorna somente a melhor categoria para cada nível. Já nesta terceira opção, o classificador deve retornar todas a possíveis categorias para cada nível, com sua respectiva probabilidade. Com isto é possível realizar uma busca pelo melhor caminho, ou seja, o que acumula a melhor probabilidade.

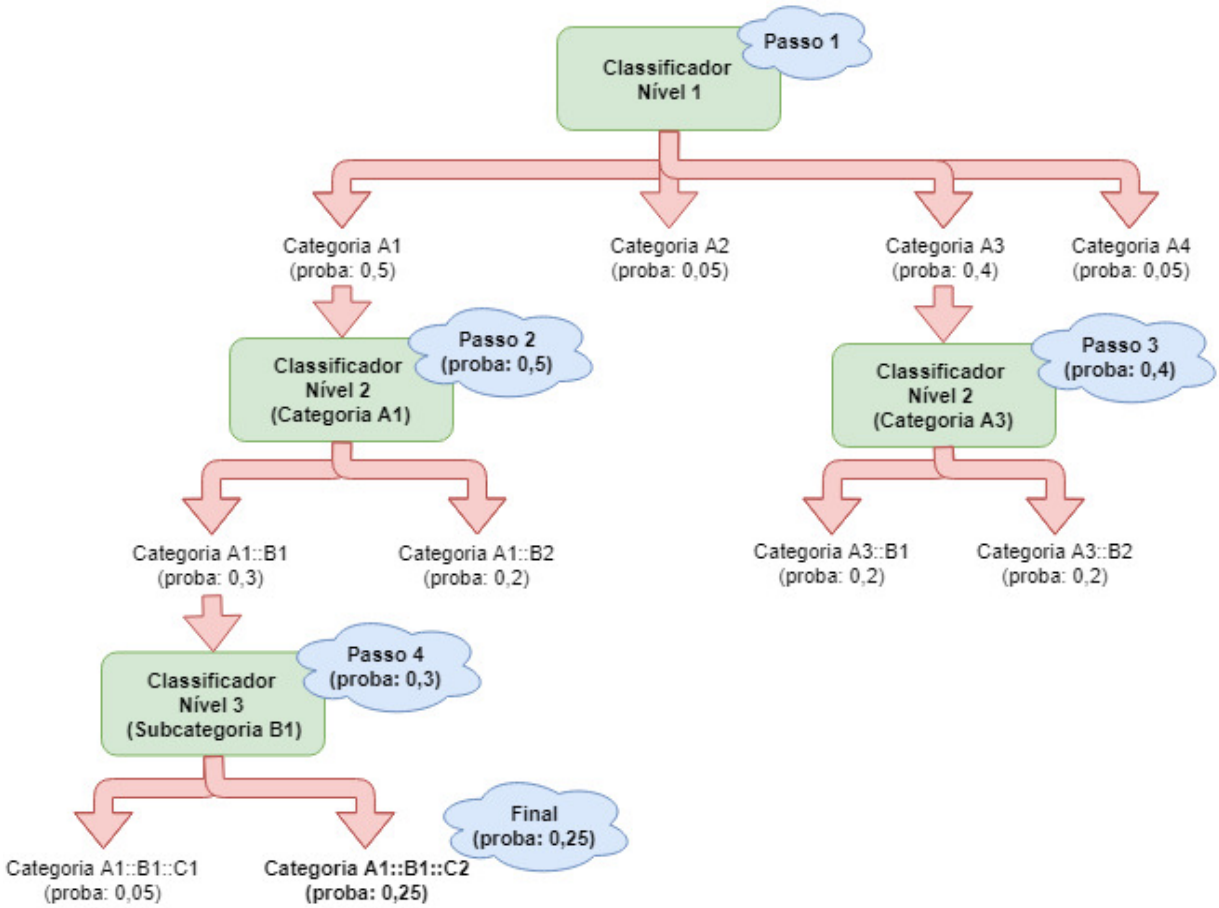

Figura 14 - Fluxo com classificadores por melhor caminho (considerando 3 níveis)

Para isto será utilizada uma implementação simplificada do algoritmo de Dijkstra[5]. A simplificação foi possível pela natureza particular do problema (árvore).

Importante observar que alguns algoritmos não retornam a probabilidade para cada classe, o que os tornam incompatíveis com este método.

As estratégias 1 e 3 serão implementadas e comparadas neste trabalho.

Métricas de avaliação

Outros trabalhos publicados relatam as acurácias ou F1-score obtidos em problemas similares.

Em “Implementing a Machine-Learning Based eCommerce Product Classification System”[1] foi relatada uma **acurácia de 97,5%**, treinando-se um algoritmo SVM com centenas de milhões de produtos.

Já “Product Classification - A Hierarchical Approach”[3] relata um **F1-score de 0,92** para vários algoritmos, treinados com aproximadamente três milhões de produtos, distribuídos em 1.628 categorias.

Em ambos os casos, o número de amostras para treinamento é muito superior ao utilizado neste trabalho.

III. Metodologia

Processamento dos Dados

Conforme as necessidades apontadas em “Exploração dos Dados”, os seguintes procedimentos foram realizados.

<u>Inversão da ordem das categorias</u>
Os níveis das categorias foram invertidos, para que fiquem do mais geral para o mais específico.

<u>Preenchimento das categorias nulas</u>
Para equalizar o número de níveis de categorias, os valores nulos foram substituídos por um valor genérico.

<u>Exclusão de níveis de categorias</u>
Os níveis 6, 7 e 8 foram excluídos, por terem pouca ou nenhuma relevância.

name	description	manufacturer	cat_1	cat_2	cat_3	cat_4	cat_5
KitchenAid - 36" Built-In Electric Cooktop - Black	Knob controls; 5 cooktop elements; 100-2700 …	KitchenAid	Appliances	Ranges, Cooktops & Ovens	Cooktops	__general__	__general__
GoPro - Camera Mount Accessory Kit - Black	Compatible with most GoPro cameras; includes ...	GoPro	Cameras & Camcorders	Camcorder Accessories	Action Camcorder Accessories	Action Camcorder Mounts	Handlebar/ Seatpost Mounts

Tabela 5 - Amostra dos dados após os ajustes

Exclusão do produto sem nome
O único produto sem nome foi excluído.

Criação de novos atributos com o caminho da categoria
Por praticidade, foram criados tributos que agregam os níveis de categoria do produto.

Exemplo:

name	Duracell - AAA Batteries (4-Pack)
cat_path_1	Connected Home & Housewares
cat_path_2	Connected Home & Housewares::Housewares
cat_path_3	Connected Home & Housewares::Housewares::Household Batteries
cat_path_4	Connected Home & Housewares::Housewares::Household Batteries::Alkaline Batteries
cat_path_5	Connected Home & Housewares::Housewares::Household Batteries::Alkaline Batteries::__general__

Onde "::" foi utilizado como separador.

Novo atributo de entrada
Por praticidade, foi criado um atributo ("all_text_features") que concatena todos os atributos de texto que serão utilizados como entrada.

Exclusão de categorias com poucos produtos
Categorias com menos de 50 produtos foram excluídas, por não apresentarem dados suficientes para o treinamento.

Este número foi definido de forma empírica, durante o treinamento. Basicamente, números menores geram muitos erros do tipo "*F-score is ill-defined and being set to 0.0 in labels with no predicted samples*". Já números maiores reduzem muito o dataset disponível.

Como comparação, vale observar que em "Product Classification - A Hierarchical Approach"[3] o mínimo considerado foi de 100 produtos.

Redução de categorias com muitos produtos
Categorias com mais de 500 produtos foram reduzidas (undersampling). Desta forma o desbalanceamento fica limitado a 10:1, e a redução das amostras otimiza o tempo de treinamento.

Implementação

Divisão dos dados
Os dados foram inicialmente divididos em um conjunto para treino (80%) e outro para a validação final (20%).

Na divisão a função ***train_test_split*** do scikit-learn foi utilizada com a opção ***stratify*** na coluna 'cat_path_5', criada no processamento e que representa a categoria completa. Assim a proporção de cada categoria foi mantida em cada um dos dois conjuntos.

Extração das features de entrada

Conforme já mencionado, foi utilizado **TF-IDF**. Mais especificamente a função ***TfidfVectorizer*** do scikit-learn, com os seguintes parâmetros:

- analyzer='word' (quebra do texto por palavras inteiras)
- lowercase=True (texto padronizado para caixa baixa)
- stop_words='english' (remoção de palavras irrelevantes)

Seleção do algoritmo

Para acelerar o grande número de treinamentos, um subconjunto de cerca de 1.400 produtos foi extraído do conjunto de treino, distribuídos em 21 categorias.

Este subconjunto foi utilizado para treinar os algoritmos candidatos. Os resultados estão exibidos na tabela abaixo.

	Tempo Treinamento (s)	F1-score Treinamento	Classificações por segundo	F1-score Validação
Naive Bayes	0.0705	0.9816	34897	0.9064
Logistic Regression	0.1400	0.9942	29015	0.9764
Random Forest	0.1390	0.9993	20470	0.9598
Decision Tree	0.1130	1.0000	31624	0.9424
SVC (linear)	0.5815	0.9964	4296	0.9800

Tabela 6 - Comparativo performance algoritmos

O classificador SVC (linear kernel) tem o melhor F1-score. Contudo é consideravelmente mais lento, tanto no treino como na predições.

O classificador Logistic Regression tem o segundo melhor F1-score, muito próximo ao primeiro, e tem um throughput significativamente maior. Assim, a escolha do classificador será realizada apena após o refinamento dos parâmetros destes dois candidatos.

Refinamento dos parâmetros (hiper-parâmetros)

Cada algoritmo de classificação possui um conjunto de parâmetros de configuração inicial, também conhecidos como hiper-parâmetros.

Os seguintes hiper-parâmetros serão refinados:

- C (ambos modelos): define a penalidade para o termo de erro no treinamento. Tem como valor padrão 1, sendo que valores menores normalizam os resultados, tornando-se mais robusto ao ruído dos dados (maior bias), valores maiores tornam o modelo mais complexo (maior variância). Ambos os modelos demonstraram maior performance com valores maiores que 1, e foram testados com valores entre 0,8 e 10.

- Penalty (Logistic Regression): define a norma utilizada para calcular a penalidade no treinamento. São duas as possibilidades: L1, que em termos simples minimiza a soma dos desvios absolutos; e L2, que minimiza o quadrado da soma dos desvios.
- Class weight (Logistic Regression): define o balanceamento das classes, podendo ser “None”, onde é uniforme, ou “balanced”, onde é inversamente proporcional a frequência das classes nos dados de treino.

Para identificar os melhores valores para o cenário em estudo foi utilizado o método ***GridSearchCV***, do scikit-learn. Neste método, definem-se possíveis valores para cada parâmetro, e é realizada uma busca exaustiva pela melhor combinação. Os resultados para os dois classificadores foram:

- SVC: 0.9828
- Logistic Regression: 0.9883

Com o refinamento o algorítmo Logistic Regression superou o SVC em F1-score, além de já possuir melhor throughput. Assim, **o algoritmo Logistic Regression será adotado**.

Os parâmetros ótimos para o Logistic Regression foram:

- ➢ Penalty = ‘l2’
- ➢ C = 5
- ➢ Class Weight: None

<u>Treinamentos Completos</u>
Após a seleção e parametrização do algoritmo, foi realizado o treinamento completo dos classificadores para as duas estratégias selecionadas em “Algoritmos e Técnicas”. Para isto, utilizou-se o conjunto completo de treinamento (80% dos dados totais). A implementação de cada classificador é descrita a seguir.

<u>Classificador Global</u>
Conforme descrito na estratégia 1, em “Algoritmos e Técnicas”, o Classificador Global não considera a estrutura hierárquica da saída. Com isto ele é simplesmente o classificador Logistic Regression refinado acima.

<u>Classificador Hierárquico com Busca</u>
Para atender o descrito na estratégia 3, foi desenvolvido um classificador chamado ***HierarchicalClassifier***, que:

- No treinamento
  - Recebe um classificador de referência (no caso o Logistic Regression refinado acima)
  - Identifica a estrutura de árvore das categorias
  - Para cada nó da árvore
    - Segmenta os dados para treinamento
    - Clona o classificador de referência e realiza seu treinamento
- Na predição
  - Determina o próximo nó da árvore, pelo algoritmo interno de busca (Dijkstra simplificado)
  - Executa a predição do classificador do nó, obtendo as probabilidades de todas as classes
  - Repete até encontrar um nó no último nível (nível 5)

Interessante apontar que, por termos ocorrencias de categorias “pai” com apenas uma categoria “filha”, foi criado um classificador de apoio chamado ***SingleClassClassifier***, que substitui o Logistic Regression nestes casos.

Refinamento

Conforme já detalhado no tópico anterior, o refinamento foi realizado durante a seleção do algoritmo de classificação.

Para o algoritmo escolhido, o F1-score subiu de 0,9764 para 0,9883.

IV. Resultados

Avaliação do Modelo e Validação

Finalmente, ambos os classificadores foram avaliados contra os 20% dos dados reservados inicialmente para este propósito.

Por serem dados nunca vistos pelos classificadores, os mesmos simulam casos reais de uso, e verificam a capacidade de generalização dos mesmos.

A próxima tabela resume os resultados obtidos.

Estratégia/ Classificador	Tempo Treinamento (s)	F1-score Treinamento	Classificações por segundo	Acurácia Validação	F1-score Validação
Global	38.5850	0.9804	32624	0.9313	0.9279
Hierarquico	6.3634	0.9835	764	0.9219	0.9267

Tabela 7 - Resultados de performance

Surpreendentemente ambos os classificadores apresentaram F1-score’s muito próximos, com uma diferença de apenas 0,0012. A diferença é um pouco maior na acurácia, de 0,0094, sempre a favor do classificador Global.

Com relação aos tempos, o classificador Global se mostrou significativamente mais lento no treinamento. Já nas classificações se mostrou muito mais veloz.

De forma geral ambos os modelos demonstram boa capacidade em prever as categorias, ficando com uma assertividade (acurácia) superior a 92%.

Para testar a robustez do modelo, o mesmo foi treinado novamente excluindo-se aleatoriamente 10% das palavras de entrada. Os testes resultaram em um F1-score de 0.9137, o que representa uma piora de apenas 1,5%. Isto comprova que o modelo é robusto para pequenas variações na entrada.

Justificativa

Conforme descrito em “Métricas de avaliação”, temos como referência uma acurácia de 0,975[1], o que é um valor excelente. O valor máximo de 0,931 obtido neste estudo é bom, mas significativamente menor.

Importante considerar que o trabalho mencionado utilizou uma quantidade de dados para treinamento várias ordens de grandeza maior, o que sabe-se ser um fator determinante para um bom score.

Outro estudo[3] reportou um F1-score de 0,92, o que está muito próximo (e abaixo) do valor máximo de 0,928 obtido neste trabalho.

Neste segundo estudo também utilizou-se uma maior quantidade de dados, mas em menor escala. Isto pode ter sido contrabalanceado com o maior número de categorias, resultando em um score similar.

V. Conclusão

Visualização Livre

O gráfico ao lado demonstra o F1-score do classificador hierárquico por nível de classificação. Nele é possível visualizar a queda do score com o aumento dos níveis, como era esperado.

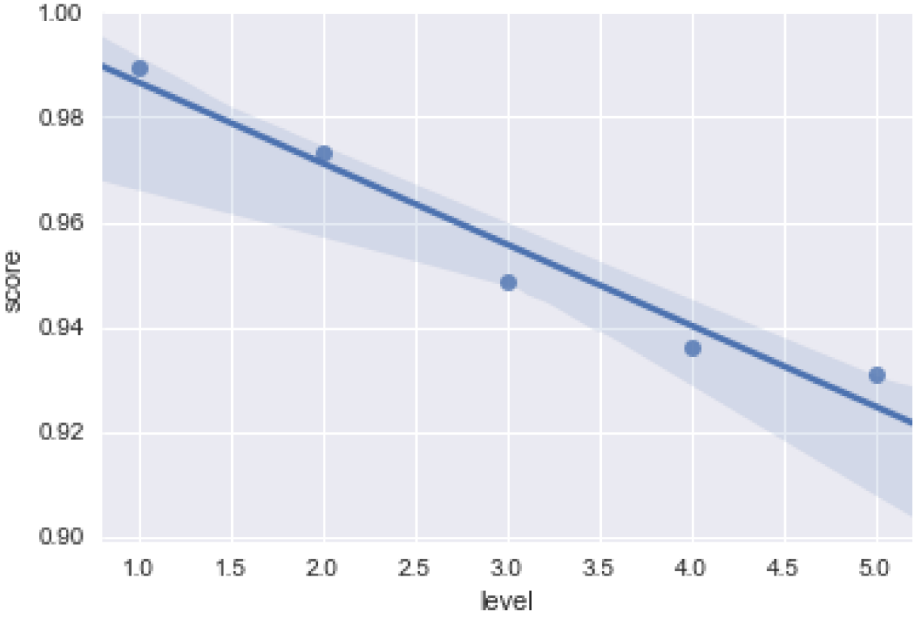

Figura 15 (ao lado) - F1-Score por quantidade de níveis na categoria

Interessante notar a forte queda entre os níveis 2 e 3, em decorrência da fragmentação das categorias. Esta fragmentação pode ser visualizada na figura 8.

Já entre os níveis 4 e 5 nota-se uma desaceleração da queda do score, visto que apenas 12% dos produtos efetivamente possuem 5 níveis. Ou seja, em 88% dos casos o classificador não precisou tomar uma decisão no nível 5.

Reflexão

Em retrospectiva, o problema de classificação foi resolvido com os seguintes passos:

1. Tratamento dos dados
2. Levantamento de algoritmos de classificação candidatos

3. Escolha de um algoritmo de classificação
4. Refinamento dos parâmetros do algoritmo
5. Criação de duas estratégias para aplicação do algoritmo
6. Validação de ambas as estratégias

A criação da estratégia hierárquica foi o passo de maior complexidade no trabalho, pois envolveu a codificação dos passos já relatados em “Implementação”.

Como pontos de melhoria no processo, pode-se citar que os algoritmos foram comparados sem otimização prévia de seus parâmetros, o que pode influenciar negativamente o resultado de alguns candidatos.

Em outro ponto similar, os parâmetros foram refinados sem considerar cada estratégia de forma individual. Visto que cada estratégia possui uma quantidade significativamente diferente de classes (vide figuras 11 e 13), o refinamento não é ótimo para nenhum dos dois cenários.

De qualquer forma, as melhorias mencionadas não foram implementadas pelo custo elevado de tempo e processamento.

Melhorias

O classificador hierárquico abre espaço para algumas melhorias, o que pode torná-lo mais atrativo que o classificador global, mesmo tendo performance ligeiramente inferior.

Podemos citar:

- Partindo de um modelo treinado, implementação de um treino parcial para novos dados. Isto é possível pois apenas os nós do qual o dado faz parte precisam ser treinados novamente.
- A mesma melhoria também é válida para novas categorias.
- Predição com níveis limitados por um threshold de confiança. Isto é, a busca na árvore é interrompida caso a probabilidade caia abaixo de um valor pré-definido, retornando uma classificação parcial.

VI. Referências

(1) Ankush Bhalotia. Implementing a Machine-Learning Based eCommerce Product Classification System. Acessado em 06/03/2018.
https://blog.dataweave.com/implementing-a-machine-learning-based-ecommerce-product-classification-system-f846d894148b

(2) Amadeus Magrabi. Boosting Product Categorization with Machine Learning. Acessado em 06/03/2018.
https://techblog.commercetools.com/boosting-product-categorization-with-machine-learning-ad4dbd30b0e8

(3) Mikael Karlsson, Anton Karlstedt. Product Classification - A Hierarchical Approach. Acessado em 06/03/2018.
http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8889613&fileOId=8889614

(4) Scikit-learn. Working With Text Data. Acessado em 06/03/2018.
http://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

(5) Dijkstra's algorithm. Acessado em 13/03/2018.
https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

(6) Best Buy API Playground. Acessado em 07/04/2018.
https://github.com/BestBuy/api-playground