Trabalho prático de MDLE - Fase 1

Instituto Superior de Engenharia de Lisboa
Mineração de Dados em Larga Escala
17 de abril de 2023
Grupo 08:
Gonçalo Fonseca - A50185
Pedro Diogo - A47573

I. Introdução

O conjunto de dados "*Influenza Outbreak Twitter Data*" [1] disponibilizado por Liang Zhao contém uma coleção de tweets relacionados com surtos de influenza feitos entre o início de 2011 e final de 2014. O problema concreto de mineração de dados a ser resolvido com este conjunto de dados pode ser a identificação de padrões nos dados que possam ser úteis para prever ou rastrear surtos de influenza em determinados locais.

A mineração de dados é o processo de descobrir informações úteis a partir de grandes quantidades de dados. Isso envolve a aplicação de várias técnicas, como agrupamento, classificação, mineração de regras de associação e deteção de anomalias para extrair *insights* e conhecimentos dos dados.

No contexto do "*Influenza Outbreak Twitter Data*", a tarefa de mineração de dados pode envolver a análise do texto dos tweets para identificar temas comuns, como os sintomas associados à influenza, as regiões onde ocorrem surtos e a gravidade dos mesmos. Os resultados da análise poderiam então ser usados para desenvolver modelos preditivos que ajudassem os funcionários de saúde pública a prepararem-se e responder melhor a surtos de influenza.

II. Análise do Dataset

Para este trabalho foi fornecido um ficheiro zip que, por sua vez, contém os seguintes ficheiros:

- **flu_locations.txt** - trata-se de uma lista que representa 48 estados norte americanos que estão associados a surtos de gripe. Este ficheiro é utilizado para identificar tweets que mencionem locais específicos onde estão a ocorrer surtos de gripe. Cada entrada no ficheiro corresponde a um estado;
- **flu_keywords.txt** - trata-se de uma lista que representa as 525 palavras-chave (atributos) que são normalmente associadas a surtos de gripe. Este ficheiro é utilizado para identificar tweets que contenham palavras ou frases específicas relacionadas com a gripe, tais como *"flu"*, *"season"*, *"caught"*, etc. Cada entrada no ficheiro corresponde a uma palavra-chave;
- **train_labels_x.csv** - contém dados de treino relativos às etiquetas utilizadas nas diferentes instâncias do conjunto de dados "x" que corresponde a um dos estados. A ocorrência de surto de gripe para o estado na próxima semana, é denotado com zero se não houver um evento na próxima semana; ou um, caso contrário;
- **train_data_x.csv** - contém dados de treino relativos aos atributos que estão presentes nas diferentes instâncias do conjunto de dados "x" que corresponde a um dos estados. A presença de uma *keyword* na instância é denotada com um número inteiro que corresponde ao número de vezes que a palavra-chave apareceu naquele tweet;
- **test_labels_x.csv** - o mesmo que train_labels_x.csv mas para dados de teste;
- **test_data_x.csv** - o mesmo que train_data_x.csv mas para dados de teste.

É importante também referir que cada ficheiro de dados (tanto os ficheiros train_data_x.csv e test_data_x.csv contêm **545 atributos** e **1095 instâncias**.

Uma característica relevante é a ausência de valores omissos (*missing values*), garantindo que o conjunto de dados esteja completo e possa ser analisado sem lacunas nos dados. Além disso, o conjunto de dados é de grande dimensão, contendo um elevado número de instâncias e atributos.

III. Pré-Processamento

Um dos problemas identificados ao analisar os conjuntos de dados foi a discrepância entre o número de palavras-chave (total de 525) no arquivo flu_keywords.txt e o número de atributos existentes nos arquivos de dados (total de 545). De acordo com [2], conseguimos perceber que os 20 atributos a mais são classificados como atributos dinâmicos, isto é, são atributos que evoluem dinamicamente. Num determinado instante do tempo podem significar algo, e no futuro podem ter um sentido completamente diferente. No contexto do problema, existem expressões utilizadas em tweets como é o caso dos *hashtags* que são caracterizados como atributos dinâmicos por possuírem este tipo de comportamento.

O único pré-processamento feito sobre os dados, foi a junção dos nomes dos atributos aos dados nas *data frames* do R. Para tal foi desenvolvida a função "join". Os atributos dinâmicos ficaram com o formato "dynamic_X", onde X representa o índice da coluna que não possuía o nome do atributo. Dessa forma, foi possível garantir a consistência dos dados e a correta junção dos conjuntos de dados, mesmo com

a discrepância inicial no número de palavras-chave e atributos nos arquivos de dados.

IV. Redução da Dimensionalidade e de Representação dos Dados

O grande objetivo e desafio desta fase do trabalho é reduzir a dimensão do conjunto de dados para fugir à *curse of dimensionality*. Para concretizar essa ideia é necessário aplicar técnicas adequadas, das quais nos vamos focar em *Feature Selection* (FS) e *Feature Reduction* (FR).

Começando com FS, utilizou-se a **variância** e a **diferença da média das medianas** para determinar quais os atributos mais relevantes. Definiram-se 3 valores de *threshold* (0.9, 0.95 e 0.99) e realizou-se a soma cumulativa dos valores de variância e diferença da média das medianas dos atributos do dataset até que o resultado corresponda aos diferentes valores de *threshold*.

A Tabela I apresenta o número de características para cada *threshold* obtido através da variância e diferença da média das medianas.

Tabela I – Tabela com o número de atributos por valor de *threshold*

	0.9	**0.95**	**0.99**
Variance	91	167	297
MM_Diff	104	181	302

Para além disso, as Figuras 1 e 2 ilustram a relevância dada pela variância e pela diferença da média das medianas dos atributos.

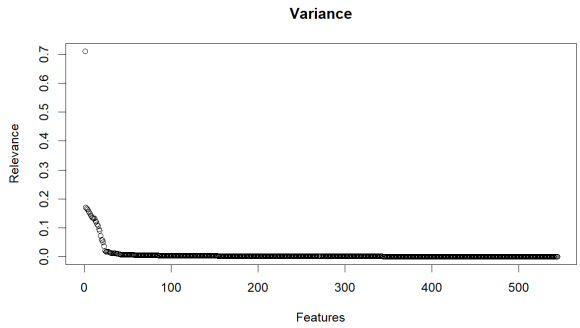

Figura 1 – Relevância dada pela variância.

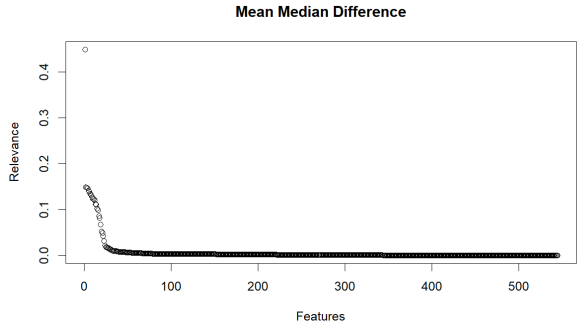

Figura 2 – Relevância dada pela diferença da média das medianas.

Mudando o foco para a *Feature Reduction*, utilizou-se o método não supervisionado *Principal Component Analysis* (PCA) que tem como objetivo determinar as componentes principais de um conjunto de dados. Começou-se por construir uma função calculate_pca que recebe o conjunto de dados, no qual se aplica a função prcomp do R [3] que faz a análise das componentes principais dos dados. Esta função retorna vários parâmetros, destacando-se o **sdev** que, por sua vez, representa os valores do desvio padrão das componentes principais da matriz de dados. Esta medida é importante pois é utilizado como uma medida da variabilidade ou dispersão dos dados ao longo de cada componente principal (**PC**). Neste caso, podemos usar os valores de **sdev** para determinar quais **PCs** manter e quais descartar. **PCs** com baixos valores de **sdev** podem não contribuir significativamente para a variabilidade dos dados e podem ser descartados, reduzindo assim o número de *features* do conjunto de dados. [4]

A Figura 3 representa o valor do desvio padrão das componentes principais do dataset.

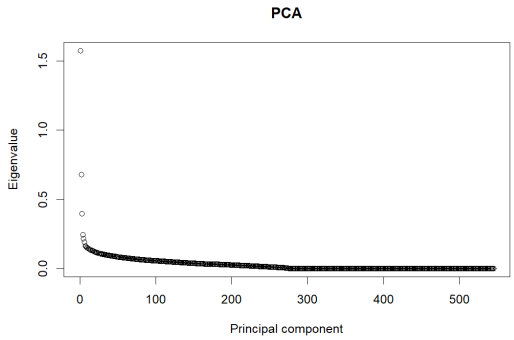

Figura 3 – Valores de desvio padrão das componentes principais.

De seguida, usando estes valores de desvio padrão, calculou-se o número de *features* onde se obteve:

Tabela II – Tabela com o número de *features* por threshold

	0.9	**0.95**	**0.99**
Variance	91	167	297
MM_Diff	104	181	302
PCA	178	210	250

Posteriormente, passou-se para a técnica de SVD. Tal como na técnica anterior, criou-se uma função **calculate_svd** que recebe os o conjunto de dados, e de seguida recorre à função **svd** do R (usada para calcular a decomposição em valores singulares de uma matriz). São retornados 3 valores, sendo uma deles o vetor **d** que contém os *singular values* do dataset. Para além disso são retornadas duas matrizes **u** e **v** que podem ser multiplicadas em conjunto com o vetor d para gerar um novo dataset reduzido.[5]

Usando os valores da variável d, gerou-se o gráfico da Figura 4.

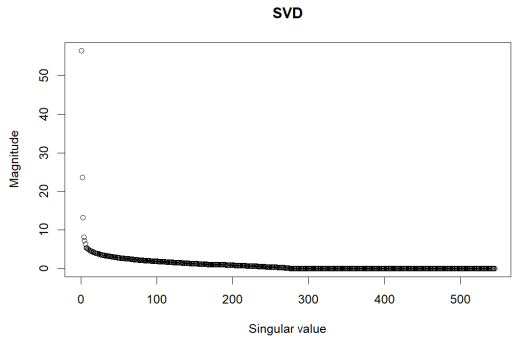

Figura 4 – Plot PCA

A tabela abaixo representa o número de *features* obtidas usando o produto de matrizes entre os valores returnados da função svd do R.

Tabela III – Tabela com o número de *features* por threshold

	0.9	**0.95**	**0.99**
Variance	91	167	297
MM_Diff	104	181	302
PCA	178	210	250
SVD	178	209	249

V. Análise dos Resultados

Para verificar quais as diferenças entre as várias técnicas utilizadas no capítulo anterior, pode-se começar por fazer uma análise à Tabela III.

Comparando os valores obtidos entre a "Variance" e o "mm_dif", podemos observar que apesar dos resultados terem sido semelhantes, o número de *features* selecionadas é maior no "mm_diff" quando comparado com a "Variance". Isso indica que o critério da Variance é mais restrito no seu critério de seleção de *features*, resultando num menor número de atributos selecionadas.

Observando os resultados do PCA e SVD, é possível observar que este têm valores bastante próximos. Isto acontece visto que ambos são técnicas diferentes, mas que compartilham semelhanças na sua abordagem. Ambos visam reduzir a dimensionalidade dos dados, identificando componentes principais ou valores singulares que capturem a variabilidade dos dados. No entanto, devido às diferenças na forma como os cálculos são realizados, é possível que os valores obtidos para o PCA e o SVD sejam ligeiramente diferentes. Portanto, é normal observar pequenas diferenças nos resultados entre estas duas técnicas, mesmo que sejam geralmente bastante semelhantes.

Por fim, comparando todas as técnicas, é possivel notar que o número de *features* aumenta nos threshold de 0.9 e 0.95. No entanto, isto não acontece com o threshold a 0.99, em que o número de *features* na variância é maior. Isso ocorre porque a variância considera a variabilidade das *features* individuais. Por outro lado, a PCA e SVD são técnicas de redução de dimensionalidade que combinam *features* e geram novas *features*, podendo resultar num menor número de recursos selecionados, mas que ainda capturam uma proporção significativa da variabilidade dos dados.

Tabela IV – Tabela com o top 10 atributos

Atributo
flu
swine
stomach
symptoms
virus
bug
strep
season
influenza
fever

A Tabela IV contém os 10 atributos que consideramos que devem ser sempre considerados quando for feita uma análise do dataset, pois apresentam a maior relevância.

VI. Conclusões

Apesar da sua elevada dimensão, o conjunto "Influenza Outbreak Twitter Data" pode ser reduzido de forma a que seja mantida grande parte da informação. Para tal é necessário aplicar técnicas de *feature selection* e *feature reduction*. Técnicas como a relevância dada pela variância e pela diferença da média das medianas provaram reduzir o dataset em aproximadamente 45%, mantendo 99% da relevância. Já técnicas como o PCA e o SVD que neste estudo tiveram resultados muito semelhantes conseguimos obter reduções de aproximadamente 55% mantendo 99% da variância.

Numa próxima fase espera-se aplicar a redução de instâncias ao dataset e o desenvolvimento de modelos para averiguar o resultado das reduções feitas tanto em instâncias como em atributos.

Referências

[1] UCI Machine Learning Repository: Influenza outbreak event prediction via Twitter data Data Set - [Em linha] [Consult. 14 abr. 2023]. Disponível em https://archive.ics.uci.edu/ml/datasets/Influenza+outbreak+event+prediction+via+Twitter+data.

[2] Zhao, L., Sun, Q., Ye, J., Chen, F., Lu, C. & Ramakrishnan, N. Feature Constrained Multi-Task Learning Models for Spatiotemporal Event Forecasting. *IEEE Transactions On Knowledge And Data Engineering*. **29**, 1059-1072 (2017)

[3] prcomp function - RDocumentation - [Em linha] [Consult. 12 abr. 2023]. Disponível em https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prcomp.

[4] GUPTA, R. - All about the Calculation Involved behind PCAGeek Culture, 1 jan. 2023. [Consult. 14 abr. 2023]. Disponível em https://medium.com/geekculture/all-about-the-calculation-involved-behind-pca-afdc5f843864.

[5] svd function - RDocumentation - [Em linha] [Consult. 12 abr. 2023]. Disponível em https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/svd.