Trabalho prático de MDLE - Fase 2

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

I. Introdução

No contexto do "*Influenza Outbreak Twitter Data*", o objetivo desta fase é resolver o problema de mineração de dados identificado no trabalho anterior, por meio da aplicação correta de técnicas de modelação e de amostragem na seleção de instâncias. De forma a lidar com problemas de representatividade das classes serão aplicadas técnicas de manipulação de instâncias, como *undersampling*, *oversampling* e *BL-SMOTE*. Para além disso, iremos explorar algumas funcionalidades que o Spark disponibiliza para dar suporte ao processamento de grandes quantidades de dados.

II. Visualização dos dados

A visualização dos dados é a prática de traduzir a informação para um contexto visual, em que neste caso serve para se entender o tipo de dados que temos, e como estes estão organizados.[1]

Os resultados apresentados são referentes às dimensões dos conjuntos de dados de treino e teste, bem como a contagem das classes presentes em ambos. O conjunto de treino tem 1095 instâncias e 545 atributos, enquanto que o conjunto de teste tem 485 instâncias e 545 atributos. Além disso, o conjunto de treino tem 49 instâncias da classe positiva e 1046 instâncias da classe negativa. Já o conjunto de teste tem 21 instâncias da classe positiva e 464 instâncias da classe negativa. Estas informações são úteis para entender a distribuição dos dados e poderão ajudar a tomar decisões sobre o pré-processamento e durante a geração do modelo de aprendizagem.

III. Seleção de atributos

Nesta secção será aplicada uma técnica de redução de atributos, recorrendo ao trabalho desenvolvido na fase anterior. De entre as várias técnicas exploradas anteriormente, optámos por escolher o SVD (*Singular Value Decomposition*). A Tabela I apresenta o número de atributos obtidos para os dois conjuntos de dados (treino e teste) recorrendo ao SVD com vários valores de *threshold*.

Tabela I – Número de atributos por *threshold* usando o SVD.

	0.9	**0.95**	**0.99**
Treino	178	209	249
Teste	85	97	112

Para o restante trabalho, os dados foram reduzidos utilizando o *threshold* de 0.99.

IV. Manipulação de Instâncias

A manipulação de instâncias refere-se ao processo de seleção de um subconjunto representativo de dados que pode substituir o conjunto de dados original, mas que continua a fornecer informações suficientes para resolver um determinado problema [2]. Esta é uma etapa importante no processo de extração de dados, uma vez que ajuda a garantir ou melhorar o desempenho [3]. Para tal, iremos explorar várias técnicas como o *Undersampling*, *Oversampling* e *BL-SMOTE*.

A. Undersampling

O *undersampling* é uma técnica de pré-processamento de dados que visa lidar com o desbalanceamento de classes em um conjunto de dados. Esse problema surge quando uma classe minoritária possui uma quantidade muito menor de exemplos do que a classe maioritária. Esta técnica consiste em reduzir a quantidade de exemplos da classe maioritária para que ela fique mais equilibrada em relação à classe minoritária. Isso pode ser feito selecionando aleatoriamente um subconjunto dos exemplos da classe maioritária ou por meio de algoritmos mais sofisticados que levam em consideração as características dos dados. O objetivo é melhorar o desempenho dos modelos de aprendizado que tendem a ser enviesados em direção à classe maioritária quando o desbalanceamento é muito grande. [4]

Ao aplicar esta técnica nos dados de treino, foram obtidas 49 instâncias para a classe positiva, e a classe negativa passou para 44 instâncias.

B. Oversampling

O *oversampling* é também uma técnica de pré-processamento de dados que visa lidar com o desbalanceamento de classes em um conjunto de dados, porém quando comparado ao *undersampling*, o *oversampling* consiste em aumentar a quantidade de exemplos da classe minoritária para equilibrar os dados. O objetivo é melhorar a representatividade da classe minoritária e evitar o enviesamento dos modelos de aprendizado em direção à classe maioritária. O *oversampling* pode ser útil em situações em que a classe minoritária é considerada importante e seu desempenho é crucial, como em diagnósticos médicos. No entanto, é importante ter cuidado para não gerar *overfitting* ao modelo ao replicar ou sintetizar muitos exemplos da classe minoritária. [5]

O resultado que se obteve foi 1047 instâncias para a classe positiva, e 1046 instâncias para a classe negativa.

C. BL-SMOTE

A última técnica de amostragem para lidar com o desequilíbrio dos dados utilizada foi o *Boderline-SMOTE*. Este algoritmo cria dados sintéticos. A função BLSMOTE da *package* smotefamily [6] começa por classificar as observações da classe minoritária em 3 grupos: SAFE/DANGER/NOISE. A classificação olha para o número de vizinhos da classe maioritária para determinar em que grupo se enquadra a observação. Apenas observações classificadas como "DANGER"são usadas para gerar instâncias sintéticas.

A função recebe os dados em formato de *dataframe* do R, sem as etiquetas. Estas são passadas como o segundo parâmetro da função. Para além destes dois argumentos, é passado um K que representa número de vizinhos mais próximos durante o processo de amostragem, um C que se trata do número de vizinhos mais próximos durante o cálculo do processo de nível seguro e por fim é passado um *method* que pode ser "type1" ou "type2". Estes métodos encontram-se descritos em [7].

Em relação aos resultados obtidos, o número de classes negativas e positivas foi idêntico ao *oversampling*, com a diferença de que neste caso os dados são sintéticos.

V. Classificação dos dados

A classificação de dados tem como objetivo separar dados em diferentes classes. Permite organizar conjuntos de dados de todos os tipos, incluindo conjuntos de dados complexos e de grandes dimensões, bem como conjuntos de dados pequenos e simples. Envolve principalmente a utilização de algoritmos que podem ser parametrizados conforme o objetivo para melhorar da classificação. [8]

Neste trabalho foram utilizados os algoritmos de classificação supervisionados SVD (*Singular Value Decomposition*) e *Random Forest* sobre os dados para cada uma das técnicas acima de *instance reduction*. Na Tabela II encontram-se os valores das métricas obtidas dos modelos, com diferentes técnicas de *sampling*. Podemos concluir que no geral o SVD conseguiu obter melhores métricas quando comparado com o *Random Forest* independentemente da técnica de amostragem de instâncias usada. Para além disso, o SVM obteve os mesmos resultados para classificação sem *sampling* e para a classificação que utilizou BL-SMOTE.

Modelo	**Técnica**	**False Positive Rate**	**Accuracy**	**Kappa**	**Positive Pred.**	**Negative Pred.**
SVM	Baseline	0.245	**0.988**	**0.844**	0.755	0.999
Random forest	Baseline	0.469	0.979	0.684	0.531	**1.000**
SVM	Undersample	0.204	0.952	0.571	0.796	0.959
Random forest	Undersample	**0.163**	0.730	0.151	**0.837**	0.725
SVM	Oversample	0.204	0.978	0.753	0.796	0.987
Random forest	Oversample	0.204	0.970	0.687	0.796	0.978
SVM	BL-SMOTE	0.245	**0.988**	**0.844**	0.755	0.999
Random forest	BL-SMOTE	0.306	0.977	0.719	0.694	0.990

Tabela II – Métricas dos modelos.

VI. Paralelização

Neste trabalho, existem vários pontos possíveis onde se pode recorrer à paralelização, destacando-se os seguintes:

- **Pré-processamento**: dependendo da complexidade dos dados e dos recursos disponíveis, diferentes etapas de pré-processamento podem ser paralelizadas. No caso do nosso trabalho, utilizamos a função *lapply* do R que permite paralelização durante a leitura e concatenação dos dados e etiquetas.
- **Redução de atributos**: apesar de ser possível paralelizar esta etapa, neste trabalho não foi feito visto que foi utilizado código da fase anterior em que não foi desenvolvido qualquer mecanismo de paralelização.
- **Geração de modelos**: este processo envolve muitas iterações e pode ser computacionalmente caro. No caso do trabalho desenvolvido não foi utilizada paralelização sobre a geração de modelos, porém uma possível abordagem seria dividir o conjunto de dados em várias partições e distribuí-las em nós "subSVM". Estes nós passariam os resultados que obtiveram para os seguintes até que

o nó final obtenha um resultado. Cada "subSVM"pode ser visto como um filtro em que apenas deixa passar instâncias que são relevantes.

VII. Conclusão

Com base no que foi realizado nesta fase, conclui-se que o objetivo foi alcançado com êxito, uma vez que foram avaliadas várias técnicas de manipulação de instâncias e testadas com vários modelos, a fim de obter a melhor técnica para este conjunto de dados.

A técnica que apresentou melhores resultados foi o BL-SMOTE, embora seja importante notar que a classificação sem *sampling* também obteve bons resultados. Contudo, caso fossem utilizados outros modelos para comparação, poderiam ter sido obtidos resultados diferentes.

Apesar de não ter sido feita muita paralelização durante o processamento dos dados, facilmente se percebe a sua importância visto que para esta quantidade de dados, o tempo de obtenção, pré-processamento, redução de instâncias, amostragem, treino e teste de modelos já foi considerável. Será sempre interessante quando trabalhamos com grandes conjuntos de dados ter uma *pipeline* bem definida que vise a redução o tempo necessário para retirar conclusões dos dados.

Na próxima fase, serão aplicadas técnicas de validação dos modelos para avaliar a eficácia do modelo, entre outros aspetos. Com a utilização dessas técnicas, será possível garantir que os modelos construídos são robustos e confiáveis.

Referências

[1] What Is Data Visualization? Definition, Examples, And Learning Resources - [Em linha] [Consult. 7 mai. 2023]. Disponível em https://www.tableau.com/learn/articles/data-visualization

[2] LIU, Huan; MOTODA, Hiroshi - Instance Selection and Construction for Data Mining. ISBN 9781441948618.

[3] Data pre-processing - Em Wikipedia [Em linha] [Consult. 7 mai. 2023]. Disponível em https://en.wikipedia.org/w/index.php?title=Data_pre-processing&oldid=1138293751

[4] Undersampling - CORP-MIDS1 (MDS), [s.d.]. [Consult. 7 mai. 2023]. Disponível em https://www.mastersindatascience.org/learning/statistics-data-science/undersampling/

[5] ADMIN - What is oversampling in data mining? [Em linha], atual. 16 fev. 2023. [Consult. 7 mai. 2023]. Disponível em https://aiblog.co.za/ai-faq/what-is-oversampling-in-data-mining

[6] A Collection of Oversampling Techniques for Class Imbalance Problem Based on SMOTE - [Em linha] [Consult. 21 abr. 2023]. Disponível em https://cran.r-project.org/web/packages/smotefamily/smotefamily.pdf

[7] H. Han, W.-Y. Wang, e B.-H. Mao, ≪Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning≫, em Advances in Intelligent Computing, Berlin, Heidelberg, 2005, pp. 878–887. doi: 10.1007/11538059_91.

[8] Classification in Data Mining Explained: Types, Classifiers & Applications [2023] - [Em linha], atual. 18 jul. 2022. [Consult. 9 mai. 2023]. Disponível em https://www.upgrad.com/blog/classification-in-data-mining/.