Trabalho Prático — Fase 3

2nd semester, 2022/2023

Duarte Valente
A47657
Grupo04
ISEL, MDLE

João Valido
A51090
Grupo04
ISEL, MDLE

I. Introdução

Este trabalho tinha como objetivo demonstrar de forma clara que compreendemos o problema da fase de modelação a resolver, aplicando correctamente técnicas de validação dos modelos de Aprendizagem Automática e avaliando, de forma critica, os resultados obtidos.

A. Âmbito

Este relatório insere-se no âmbito da realização da fase 3 do trabalho prático, continuação da fase 2, da unidade curricular de Mineração de Dados de Larga Escala, do Mestrado em Engenharia Informática e Multimédia do DEETC do ISEL.

B. Documentação

Documentação de apoio à unidade curricular de Mineração de Dados de Larga Escala.

II. Desenvolvimento

A. Problemas e soluções na fase de modulação

A modelação é uma fase crucial na mineração de dados de larga escala, pois é nessa fase que os dados brutos são transformados em modelos que podem ser usados para prever, classificar ou identificar padrões nos dados. No entanto, existem vários problemas que podem surgir durante a modelação de dados de larga escala. Aqui estão alguns exemplos desses problemas e possíveis soluções:

- Problema: Overfitting - Este problema ocorre quando o modelo se ajusta demais aos dados de treinamento e não é capaz de generalizar bem para novos dados.
- Solução: Uma solução comum para o overfitting é usar técnicas de regularização, como a regularização L1 ou L2, ou aumentar a quantidade de dados de treino para ajudar a suavizar o modelo.
- Problema: Dimensionalidade elevada - À medida que o número de variáveis aumenta, a dimensionalidade dos dados aumenta, tornando o modelo mais complexo e difícil de interpretar.
- Solução: Para lidar com a dimensionalidade elevada, pode-se aplicar técnicas de seleção de características para reduzir o número de variáveis ou técnicas de redução de dimensionalidade, como Análise de Componentes Principais (PCA).
- Problema: Dados desbalanceados - Em alguns casos, a distribuição dos dados pode ser desbalanceada, com uma classe ou categoria dominando os dados.
- Solução: Para lidar com dados desbalanceados, podem-se aplicar técnicas de amostragem, como a undersampling ou oversampling.

B. Paralelização de ações

Uma vantagem da utilização de R e Spark é ser possível paralelizar ações utilizando as capacidades de computação paralela fornecidas pelo Spark. Pois o Spark foi projetado para lidar com tarefas de processamento de dados em larga escala de forma distribuída e paralela. Assim, para o desenvolvimento desta fase do projeto, como foi necessário processar os dados de forma a extrair conhecimento para um determinado modelo aprender a prever um certo resultado com base em características extraídas do dataset, aproveitamos este processamento paralelo do Spark para acelerar este processo computacional. Para este efeito, foram criados dataframes de Spark na sessão Spark, para onde foram passados os dados pretendidos, efetuando assim o processamento com os mesmos.

C. Feature Selection

- Antes de repartirmos os dados para treino e teste, temos de selecionar as features mais relevantes deste dataset, que por sua vez já foram calculadas na fase anterior. Assim, selecionamos as 48 features mais relevantes das 545 features presentes neste dataset.

D. Split do dataset

- Para efeitos de treino, os dados, foram divididos em treino e teste, sendo os dados de treino 80% do dataset e os de teste os restantes 20%. Neste processo também foi verificado o número de instâncias de cada classe presente no dataset de treino e teste, verificando que para o dataset de treino obtivémos 4224 instâncias da classe 0 e 170 instâncias da classe 1. Enquanto para o data set de teste, verificámos que tínhamos 1038 instâncias da classe 0 e 43 da classe 1.

E. Treino do modelo sem aplicação de técnicas de amostragem de seleção de instâncias

- Para uma aprendizagem sobre os dados foi criado um modelo de classificação através da função ml_random_forest. Modelo esse que depois de treinado, foram efetuadas previsões para os dados de teste, podendo assim efetuar uma medição sobre os seus desempenhos. Os resultados mostraram que a precisão do modelo foi de 96% e que a taxa de falsos positivos foi de 0.933.

F. Aplicação de técnicas de amostragem de seleção de instâncias

Para uma melhor compreensão dos dados, foram executadas duas técnicas de amostragem de seleção de instâncias:

- Undersampling: Passa por remover instâncias da classe majoritária, equilibrando a distribuição das classes no conjunto de dados. O que ajuda o processo de aprendizagem pois esta falta de balanceamento de classes afeta o modelo e induz o mesmo a favorecer a classe majoritária, resultando em um desempenho inferior na classificação das classes minoritárias.
- Oversampling: Por outro lado, o oversampling, passa por fazer o oposto do undersampling que em vez de diminuir o número de instâncias na classe majoritária, este replica ou gera sinteticamente instâncias na classe minoritária. O que pelas mesmas razões do undersampling tambem ajuda no processo de aprendizagem do modelo, uma vez que a distribuição do número de classes fica mais equilibrada.

Para o processo de undersampling o modelo apresentou uma procissão de 89%, no entanto a taxa de falsos positivos foi de 0.535. Já para o processo de oversampling o modelo apresentou uma procissão de 93%, e a taxa de falsos positivos tambem de 0.535.

G. Validação

A validação do modelo Random Forest utilizado, é uma etapa crítica para posteriormente podermos efetuar uma avaliação do desempenho do modelo. Assim, existem várias técnicas de validação que podem ser utilizadas para avaliar esta precisão e o desempenho. Duas dessas técnicas populares são por exemplo a validação cruzada e a divisão dos dados em treino e teste. Validação cruzada: A técnica de validação cruzada é popularmente utilizada na validação de modelos, e ainda mais frequentemente quando os dados disponíveis são limitados. Essencialmente, esta técnica envolve a divisão do conjunto de dados em k partes separadas, referidas como 'folds', treinando assim o modelo em k-1 partes e depois testado na restante. A seguir, o processo é realizado k vezes no total, com rotação entre as partes utilizadas tanto para treinamento quanto para teste. Em R, o Spark permite a utilização da função $ml\_cross\_validation()$ para efetuar a validação cruzada dos modelos. No entanto optamos por utilizar uma outra abordagem para efetuar a validação dos dados. Assim, a abordagem utilizada neste trabalho foi de dividir o conjunto de dados em conjuntos de treino e teste. Sendo o conjunto de treino utilizado para treinar o modelo Random Forest, enquanto o conjunto de teste foi usado para avaliar o desempenho do modelo. Para efetuar esta divisão dos dados optamos por repartir os mesmos em 80% para dados de treino e os restantes 20% como dados de teste. Em R, o Spark, disponibiliza a função randomSplit() que nos permite assim, dividir o conjunto de dados em treino e teste. Estas são apenas duas técnicas de validação que podem ser aplicadas por modelos Random Forest em R. No entanto, existem outras técnicas disponíveis, como a validação (holdout) em uma única amostra, bootstrapping, etc. O importante a realçar é que a escolha da técnica depende sempre do conjunto de dados em causa, tendo de avaliar características como o tamanho do problema e necessidades específicas do conjunto

H. Avaliação

TABLE I
TABELA DE COMPARAÇÃO

	FalsePositive	Accuracy	Kappa	PosPred	NegPred
Baseline model	0.814	**0.967**	0.298	0.186	**0.999**
Undersampling	0.535	0.887	0.200	0.465	0.905
Oversampling	**0.535**	0.933	**0.324**	**0.465**	0.953

Ao comparar os resultados dos três métodos ("Baseline model", "Undersampling" e "Oversampling") na TABLE 1, podemos observar o seguinte:

- Em relação à taxa de falsos positivos (FP), os métodos "Undersampling" e "Oversampling" apresentaram resultados idênticos, com valores de 0.535. O método "Baseline model" apresentou um desempenho pior, com uma taxa de FP de 0.814. Isso significa que o modelo "Baseline" fez mais previsões incorretas de que uma observação era positiva, quando na verdade era negativa, em comparação com os outros dois métodos.
- Em termos de accuracy geral, o modelo "Baseline" novamente teve o melhor desempenho, com uma accuracy de 0.967. Isso significa que o modelo "Baseline" classificou corretamente 96,7% de todas as observações. Os métodos "Undersampling" e "Oversampling" apresentaram accuracy menores, com valores de 0.887 e 0.933, respectivamente.
- O coeficiente kappa, que mede a concordância entre as previsões do modelo e as observações reais, corrigido pela chance aleatória, variou de 0,200 (método "Undersampling") a 0,324 (método "Oversampling"). O modelo "Baseline" apresentou um valor de kappa de 0,289, que fica entre os outros dois métodos.
- Em relação às taxas de previsões positivas corretas (TPR) e negativas corretas (TNR), os métodos "Undersampling" e "Oversampling" apresentaram resultados muito semelhantes, com valores de 0,465 e 0,905 (método "Undersampling") e 0,465 e 0,953 (método "Oversampling"). O modelo "Baseline" apresentou uma TPR muito baixa

de 0,186, sugerindo que teve dificuldade em identificar observações verdadeiramente positivas, mas apresentou um TNR quase perfeito de 0,999, sugerindo que conseguiu identificar corretamente a grande maioria das observações negativas.

TABLE II
TABELA DE AVALIAÇÃO

	Baseline model	Undersampling	Oversampling
Mean Square Error	**0.03117183**	0.1364416	0.1023194
Root Mean Square Error	**0.1765555**	0.3693801	0.319874
Mean Absolute Error	**0.0566093**	0.3332846	0.2776935
Relative Absolute Error	**1.423131**	8.378619	6.981085
Correlation	**0.4303562**	0.32949	0.3046169

Ao examinar a TABLE II, podemos notar que as várias colunas apresentam as seguintes observações. O valor do Erro Quadrado Médio (MSE) determina o nível de precisão das previsões em relação aos valores reais. A abordagem "Modelo de linha de base" tem o menor valor de MSE de 0,03117183, indicando que o modelo tem o menor erro quadrático médio. Por outro lado, os métodos "Undersampling" e "Oversampling" apresentam valores de MSE relativamente mais altos, indicando erros quadráticos médios maiores nas respetivas previsões. Em termos de Root Mean Square Error (RMSE), o "Baseline model" apresenta o menor valor de 0,1765555, enquanto os modelos "Undersampling" e "Oversampling" apresentam os maiores valores de RMSE, sugerindo que seus modelos possuem um erro médio que é maior que os valores reais.

O MAE, ou Erro Absoluto Médio, é uma métrica para medir o grau de erro entre os valores projetados e os valores reais. A abordagem "Modelo de linha de base" tem o menor valor de MAE de 0,0566093, o que implica que esse método tem a menor margem de erro em comparação com os outros métodos. Os modelos "Undersampling" e "Oversampling", por outro lado, apresentam valores de MAE aumentados, indicando que o grau de erro é mais significativo.

A métrica Relative Absolute Error (RAE) é utilizada para avaliar a precisão de vários métodos. Mais uma vez, o método "Baseline model" apresenta o menor valor, registrando-se em 1,423131. Quanto aos métodos "Undersampling" e "Oversampling", seus valores de RAE são consideravelmente maiores, implicando um erro relativo absoluto maior quando comparado ao valor real.

Ao avaliar a correlação, o modelo "Baseline model" destaca-se com o valor mais alto de 0,4303562, o que destaca uma semelhança maior entre as previsões e os resultados factuais. Comparativamente, os métodos "Undersampling" e "Oversampling" apresentam valores de correlação inferiores, indicando um grau de correlação mais fraco.

No geral, a técnica "Baseline model" parece superar as abordagens "Undersampling" e "Oversampling" de acordo com as métricas obtidas.

I. Comparação com outros projetos

Em comparação com outros modelos de objetivo semelhante podemos tirar tambem algumas conclusoes. Assim, utilizamos os dados analizados e cocluidos do artigo "Early Detection of Seasonal Outbreaks from Twitter Data Using Machine Learning Approaches by Furqan Aziz" e efetuamos algumas comparações sobre o modelo Random Forest uma vez que foi este o utilizado.

Algorithm	Precision (%)	Accuracy (%)	Recall (%)
DT	86	84	87
KNN	87	85	85
SVM	87	85	85
RF	88	87	87

Fig. 1. Tabela do artigo [1].

Podemos então observar que em relação á precisão do mesmo, esta foi bastante semelhante o que aparenta ser um bom indicador uma vez que a precisão indica a capacidade do modelo em evitar classificar erradamente instâncias negativas como positivas.

Já a accuracy do modelo também se revelou ser bastante alta (96%) sendo a do artigo de 87% o que mostra que o nosso modelo calcula bastante bem um grande número de instâncias em comparação ao número total de instâncias. Esta maior accuary deve-se provavelmente ao facto de terem sido utilizadas menos amostras de teste para casos positivos uma vez que para as 4394 instâncias apenas 170 eram casos positivos.

Por fim, o recall foi o valor mais dispare, pois, o nosso modelo tem um recall de 19% enquanto o do artigo tem de 87%. Esta diferença de valores, deve-se também, ao facto da diferença significativa de casos positivos e negativos nos dados de treino. Pois para os modelos onde foi efetuado oversampling e undersampling nos dados, para balancear este número, verificamos que o recall aumentou para 47% e 51% respetivamente. Mesmo assim, este valor encontra-se longe dos 87% calculados no artigo, o que pode indicar que poderíamos ter utilizado algum método para ajudar o modelo a identificar corretamente todas as instâncias positivas presentes nos dados.

REFERENCES

[1] https://www.hindawi.com/journals/complexity/2021/5520366/.