CEFET/RJ
Bacharelado em Ciência da Computação
GCC1734 - Inteligência Artificial (2024.2)
Trabalho 04 - Aprendizado de Máquina

Prof. Eduardo Bezerra (ebezerra@cefet-rj.br)

26 de janeiro de 2025

Sumário

1 Previsão de pagamento de empréstimos (2 pts)

Uma instituição financeira (fictícia) possui uma base de dados com o histórico de crediário oferecido aos seus clientes. Baseado neste histórico, a instituição deseja investigar a criação de modelos de classificação para inferir se um novo cliente que submeteu uma requisição de empréstimo pagará ou não a dívida, caso o banco resolva realizar esse empréstimo. O objetivo é predizer se um novo cliente pagaria ou não uma dívida contraída, tendo como base as características desse novo cliente. Uma vez treinado, um modelo de classificação para esse problema poderá inferir se um novo cliente irá ou não honrar um eventual empréstimo concedido a ele.

O conjunto de dados a ser utilizado para treinamento possui 1500 exemplos, e contém dados relativos a créditos (empréstimos) concedidos aos clientes da instituição financeira. Esses registros estão contidos no arquivo `credtrain.txt`, que é fornecido juntamente com esse documento. Para cada cliente, são definidos 11 atributos (variáveis, características). Além disso, a última coluna de cada exemplo informa se o cliente honrou ou não o pagamento do empréstimo. Na Tabela 1, encontramos a descrição dos atributos.

Tabela 1: Esquema do conjunto de dados com histórico de clientes.

Variável	**Descrição**	**Tipo**	**Domínio**
ESCT	Estado civil	Categórica	0,1,2,3
NDEP	Número de dependentes	Categórica	0,1,2,3,4,5,6,7
RENDA	Renda Familiar	Numérica	300-9675
TIPOR	Tipo de residência	Categórica	0,1
VBEM	Valor do bem a ser adquirido	Numérica	300-6000
NPARC	Número de parcelas	Numérica	1-24
VPARC	Valor da parcela	Numérica	50-719
TEL	Se o cliente possui telefone	Categórica	0,1
IDADE	Idade do cliente	Numérica	18-70
RESMS	Tempo de moradia (em meses)	Numérica	0-420
ENTRADA	Valor da entrada	Numérica	0-1300
CLASSE	=1 se o cliente pagou a dívida	Categórica	0,1

Repare que esse conjunto de dados contém diversos atributos categóricos. Repare também que, dentre os atributos numéricos, há uma grande discrepância entre as suas respectivas faixas de valores. Modelos de aprendizado de máquina não podem ser treinados sobre atributos que não são numéricos. Além disso, é sabido que diferenças grandes entre as faixas de valores dos atributos atrapalha o processo de treinamento realizado por alguns algoritmos. Sendo assim, antes de iniciar o treinamento, você deve obrigatoriamente realizar dois passos de pré-processamento sobre esses dados:

- Codificação de atributos discretos. Aplique as técnicas de codificação *One-Hot encoding* ou *Ordinal Encoding*, a que for mais adequada para cada atributo discreto.

- Normalização de atributos contínuos. Aplique a técnica de normalização implementada na classe `sklearn.preprocessing.StandardScaler`.

Você deve criar modelos de classificação por meio dos algoritmos de aprendizado de máquina:

- `sklearn.linear_model.LogisticRegression`
- `sklearn.neighbors.KNeighborsClassifier`
- `xgboost.XGBClassifier`

Por simplicidade, você pode manter os valores *default* dos hiperparâmetros de cada algoritmo.

Após o treinamento, você deve avaliar a qualidade preditiva dos modelos gerados. Para isso, você deve usar os exemplos contidos no arquivo `credtest.txt`. Isso permitirá que você avalie o quão efetivo foi o passo de treinamento dos modelos, ou seja, qual o poder preditivo de cada modelo de classificação. Produza a *matriz de confusão* (*confusion matrix*) relativa aos resultados da fase de testes (`credtest`). Apresente também o resultado produzido pela função `classification_report` do Scikit-Learn. Produza a *matriz de confusão* (*confusion matrix*) relativa aos resultados da fase de testes (`credtest`). Apresente também o resultado produzido pela função `classification_report` do Scikit-Learn.

2 Engenharia de *Features* (2 pts)

Considere novamente o problema de classificação apresentado na parte 1 deste trabalho. Em aula, estudamos a aplicação da técnica *Target Encoding* para codificar atributos categóricos. Nesta parte, sua tarefa é aplicar essas técnicas para codificação de *features* categóricas nesse conjunto de dados. Para isso, você deve usar a biblioteca Categories Encoder[1]. Estude a documentação fornecida por essa biblioteca para produzir uma boa combinação de técnicas de codificação de atributos categóricos. Em seguida, reexecute o processo de construção dos modelos de classificação. Apresente uma análise comparativa dos resultados que você obtiver ao usar cada uma das três técnicas de codificação.

3 Conjuntos desbalanceados - parte I (2 pts)

Nesta parte do trabalho, são fornecidos arquivos no formato `npy`: `X_train.npy`, `y_train.npy`, `X_val.npy`, `y_val.npy`, `X_test.npy`, `y_test.npy`[2]. Esses arquivos correspondem a conjuntos de treino, validação e testes para um problema de classificação binária.[3] O trecho de código abaixo ilustra como é possível fazer a carga desses arquivos.

[1] `https://contrib.scikit-learn.org/category_encoders/`

[2] `https://github.com/AILAB-CEFET-RJ/cic1205/tree/main/data/falldetection`

[3] A fonte desses dados e o modo pelo qual eles foram pré-processados para geração desse conjunto são aspectos irrelevantes para o que deve ser feito neste trabalho. Contudo, se você estiver curioso sobre a fontes de dados, consulte esse link: `https://zenodo.org/records/12760391`.

```python
import numpy as np
# File name of the saved .npy file
file_name = "X_train.npy"

# Load the NumPy array from the .npy file
X_train = np.load(file_name)

# Print the loaded array
print("Loaded Array:", X_train)
```

Você vai notar ao inspecionar as matrizes `y_*` correspondem aos valores binários do atributo alvo. Você irá notar que esses conjuntos de dados são altamente desbalanceados.

Sua tarefa nesta parte do trabalho é investigar se a aplicação de alguma técnica de balanceamento de dados é efetiva no sentido de produzir um modelo que tenha maior desempenho preditivo. Ou seja, você vai comparar se um modelo treinado sem aplicar balanceamento é pior, ou melhor (do ponto de vista preditivo) do que um modelo treinado após a aplicação de alguma técnica de balanceamento. Você deve obrigatoriamente testar as três alternativas de solução descritas em aula (*undersampling*, *oversampling* e *alteração de limiar*), mas está livre para testar outras, se quiser. Faça essa investigação utilizando um único algoritmo de aprendizado, a saber, o `xgboost.XGBClassifier`. Em sua análise dos resultados para cada fonte, forneça as matrizes de confusão obtidas, assim como os relatórios de classificação obtidos por meio da função `classification_report` do Scikit-Learn.

4 Conjuntos desbalanceados - parte II (2 pts)

Nesta parte, você deve realizar um novo experimento como tentativa para mitigar o problema de desbalanceamento dos conjuntos de dados apresentados na parte 3. Entrento, desta vez, você irá resolver um problema de regressão.

O procedimento a ser usado nesse experimento é descrito a seguir. Para cada conjunto de dados fornecido, marque todos os alvos diferentes de zero como **1** e o restante como **0**. Ao fazer isso, você terá criado um conjunto de dados para classificação binária correspondente a cada conjunto de dados fornecido. Em seguida, para cada um desses conjuntos de dados binários, execute os passos listados abaixo.

a. Treine um modelo de classificação binária a partir deste conjunto de dados. Chame esse modelo de $\mathcal{C}$.

b. Treine um modelo de regressão apenas nos pontos de dados que forem classificados por $\mathcal{C}$ como sendo da classe **1**. Chame esse modelo de $\mathcal{R}$.

c. Para obter $\mathcal{R}'(\mathbf{x})$, i.e., a predição de regressão para um novo exemplo $\mathbf{x}$, inicialmente compute $\mathcal{C}(\mathbf{x})$. Em seguida, use o valor computado para computar $\mathcal{R}'(\mathbf{x})$ da seguinte

forma:

$$\mathcal{R}'(\mathbf{x}) = \begin{cases} 0 & \text{se } \mathcal{C}(\mathbf{x}) = 0 \\ \mathcal{R}(\mathbf{x}) & \text{se } \mathcal{C}(\mathbf{x}) = 1 \end{cases}$$

Você está livre para escolher o classificador e o regressor para usar na implementação do procedimento descrito acima. Reporte os resultados desse experimento para o conjunto de testes fornecido. Compare seu modelo produzido por meio desse procedimento com os modelos produzidos na parte 3. Esse procedimento produziu um melhor modelo do ponto de vista preditivo?

5 Busca de hiperparâmetros (2 pts)

Considere novamente os arquivos fornecidos na parte 3. Tomando como ponto de partida o código fornecido em aula, nessa parte do trabalho você deve realizar um experimento para encontrar uma boa combinação de hiperparâmetros para ajustar um modelo para esse conjunto de dados. Para isso, você deve usar o framework `Optuna`.

Como algoritmo de aprendizado, você deve usar o `xgboost.XGBClassifier`. Estude a documentação relativa a esse algoritmo para selecionar quais hiperparâmetros irá explorar. Você consegue obter um melhor modelo, quando comparado ao modelo gerado sem otimização de hiperparâmetros?

Especificação da entrega

- Você deve produzir um notebook Jupyter que deve apresentar as **implementações e os resultados de execução** de cada parte desse trabalho. Nesse notebook, descreva **em detalhes** de que forma implementou cada parte desse trabalho. Já é fornecido junto com este enunciado um notebook para você usar como ponto de partida neste trabalho.

- O único arquivo a ser submetido é o notebook Jypyter. Esse arquivo deve ser nomeado com o seguinte padrão: IA_T4_SEU_NOME_COMPLETO.ipynb. Um exemplo: IA_T4_EDUARDO_BEZERRA_DA_SILVA.ipynb. Siga à risca essa convenção de nomenclatura.

- Você deve também elaborar um vídeo (cuja duração aproximada foi especificada no primeiro dia de aula) no qual você deve explicar os aspectos mais importantes de cada parte do seu trabalho. Nesse vídeo, você também deve demonstrar a execução de cada parte e apresentar uma análise dos resultados obtidos. O link para acesso a esse vídeo deve estar contido na primeira célula (de texto) do notebook Jupyter.

- A entrega aqui especificada deve ser realizada pela plataforma MS Teams, até a data estabelecida. Trabalhos entregues com atraso irão sofrer desconto na nota (20% a cada dia de atraso).

- Esse trabalho é individual. Você é livre para discutir com seus colegas de turma sobre as partes desse trabalho, mas deve manter para si as suas soluções. Eventuais cópias em quaisquer partes do trabalho serão penalizadas com nota zero.