CEFET/RJ
Programa de Pós-graduação em Ciência da Computação
Aprendizado de Máquina (CIC1205) - 2023
Trabalho 1

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

28 de outubro de 2023

Sumário

1 Predição de pagamento de empréstimos (2 pts) 3

2 Predição de preços de diamantes (2 pts) 4

3 Conjuntos desbalanceados (2 pts) 5

4 Calibração de modelos (2 pts) 5

5 Busca de hiperparâmetros (2 pts) 6

1 Predição 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 que não são numéricos. Repare também que, dentre os atributos numéricos, há uma grande discrepância entre as suas respectivas faixas de valores. Modelos de redes neurais 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. Sendo assim, antes de iniciar o treinamento, é preciso realizar diversos passos de pré-processamento sobre esses dados. Esses passos já são fornecidos em um notebook Jupyter.

Você deve criar modelos de classificação por meio dos algoritmos de aprendizado de máquina implementados nas seguintes classes da biblioteca Scikit-Learn:

- sklearn.linear_model.LogisticRegression

- sklearn.tree.DecisionTreeClassifier
- sklearn.ensemble.RandomForestClassifier
- sklearn.neighbors.KNeighborsClassifier
- sklearn.ensemble.GradientBoostingClassifier

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 Predição de preços de diamantes (2 pts)

Nessa parte, você deve treinar um modelo de rede neural MLP para realizar uma tarefa de regressão sobre o conjunto de dados `Diamond`. Esse conjunto de dados (junto com uma descrição dos seus atributos) pode ser obtido em `https://www.kaggle.com/shivam2503/diamonds`. Em particular, você deve criar um modelo para predizer o valor do preço (representado na variável dependente `price`) de um diamante usando os demais atributos como variáveis independentes.

Repare que o conjunto de dados `Diamond` também contém variáveis não-numéricas. Sendo assim, você também precisará realizar passos de pré-processamento sobre o conjunto de dados antes de iniciar o treinamento do modelo. Para isso, tome como exemplo os passos de pré-processamento realizados sobre o conjunto de dados de clientes.

Você deve criar modelos de predição (regressão) de preços por meio dos algoritmos de aprendizado de máquina implementados nas seguintes classes da biblioteca Scikit-Learn:

- sklearn.linear_model.LinearRegression
- sklearn.linear_model.Lasso
- sklearn.tree.DecisionTreeRegressor
- sklearn.ensemble.RandomForestRegressor
- sklearn.neighbors.KNeighborsRegressor
- `sklearn.ensemble.GradientBoostingRegressor`

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

Após o treinamento, você deve avaliar a qualidade preditiva de cada modelo de classificação resultante. Para isso, você deve separar 20% dos exemplos fornecidos para o conjunto de teste. Isso permitirá que você avalie o quão efetivo foi o treinamento dos modelos. Certifique-se de avaliar todos os modelos sobre o mesmo conjunto de teste. Como métricas de avaliação, use o MSE e o coeficiente e determinação R^2.

3 Conjuntos desbalanceados (2 pts)

Nesta parte do trabalho, são fornecidos cinco arquivos no formato Pickle, cada um dos quais produzido a partir de uma fonte de dados diferente: A602.pickle, A621.pickle, A627.pickle, A636.pickle, A652.pickle. Cada um desses arquivos contém conjuntos de treino, validação e testes da fonte de dados correspondente.[1] O trecho de código abaixo ilustra como é possível ter acesso aos conjuntos de dados para cada fonte. Nesse trecho de código, `outfilename` é o nome de um dos cinco arquivo fornecidos.

```
import numpy as np
import pickle
file = open(outfilename, 'rb')
(X_train, y_train, X_val, y_val, X_test, y_test) = pickle.load(file)
print(f"Shapes: ", X_train.shape, X_val.shape, X_test.shape)
```

Você vai notar ao inspecionar as matrizes `y_*` de cada fonte que esses são conjuntos de dados para um problema de classificação binária. Irá notar também 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 `sklearn.ensemble.GradientBoostingRegressor`. 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 Calibração de modelos (2 pts)

Considere novamente o arquivo de uma das fontes de dados fornecido na parte 3, o A652.pickle. Considere que, no domínio do problema em questão, é importante que as probabilidades de

[1] 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, isso será alvo de estudo em aulas futuras do curso.

predição do modelo estejam corretamente calibradas. Sua tarefa nesta parte é, de início, ajustar um modelo de classificação sobre esse conjunto de dados. Para isso, use novamente o algoritmo `sklearn.ensemble.GradientBoostingRegressor`. Em seguida, investigue o grau de calibração do modelo resultante e, conforme for o caso, aplique alguma técnica para calibrar os resultados do modelo. Apresente gráficos para ilustrar os graus de calibração dos modelos antes e após aplicar a calibração. Apresente uma análise dos resultados obtidos.

5 Busca de hiperparâmetros (2 pts)

Considere novamente um dos arquivos fornecidos na parte 3, o A652.pickle. Tomando como ponto de partida o código fornecido em aula, nessa parte do trabalho você deve realizar um experimento para encontrar um boa combinação de hiperparâmetros para ajustar um modelo para esse conjunto de dados. Como algoritmo de aprendizado, você deve escolher um dos listados na parte 1. Estude a documentação do Scikit Learn relativa ao algoritmo que você escolher, para selecionar quais hiperparâmetros irá explorar. Você também é livre para escolher entre duas estratégias de busca de hiperparâmetros, *Grid Search* ou *Random Search*. Apresente uma análise dos resultados encontrados.

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: AM_T1_SEU_NOME_COMPLETO.ipynb. Um exemplo: AM_T1_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.