APRENDIZADO DE MÁQUINA (CIC1205) - 2025

Trabalho 1

Prof. Eduardo Bezerra (ebezerra@cefet-rj.br)
CEFET/RJ - Programa de Pós-graduação em Ciência da Computação

Abril/2025

Sumário

Especificação da entrega

1. Você deve produzir um único *notebook* Jupyter. Esse *notebook* 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.

2. Já é fornecido junto com este enunciado um notebook Jupyter para você usar como ponto de partida. Você encontra esse arquivo nesse link.

3. Utilize a linguagem de marcação (*markdown*) para organizar seu *notebook* em seções que reflitam cada parte deste trabalho. Sua solução para cada parte deve está facilmente encontrável nesse *notebook*.

4. 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.

5. 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.

6. Cuide para que a qualidade áudio-visual do seu vídeo esteja adequada para que seu conteúdo possa ser avaliado.

7. 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).

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

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. É também fornecido um conjunto de exemplos de teste no arquivo credtest.txt. Para cada cliente, são definidos 11 atributos (variáveis preditoras). Além disso, a última coluna de cada arquivo 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

Antes de iniciar o treinamento, é preciso realizar diversos passos de pré-processamento sobre esses dados. Alguns aspectos que você deve levar em contar na sua solução:

- Esse conjunto de dados contém diversos atributos que são categóricos. Modelos de AM não podem ser treinados no Scikit-Learn sobre atributos cujos valores são cadeias de caracteres. Sendo assim, você deve tomar providências para codificar numericamente esses atributos de maneira apropriada.
- Dentre os atributos numéricos, há uma grande discrepância entre as suas respectivas faixas de valores. É sabido que diferenças grandes entre as faixas de valores dos atributos numéricos pode atrapalhar o processo de treinamento de alguns algoritmos de AM.

Após realizar os passos de pré-processamento adequados, você deve criar modelos de classificação por meio dos algoritmos de aprendizado de máquina implementados nas seguintes classes da biblioteca Scikit-Learn. (Por simplicidade, você pode manter os valores *default* dos hiperparâmetros de cada algoritmo.)

1. `sklearn.linear_model.LogisticRegression`
2. `sklearn.neighbors.KNeighborsClassifier`
3. `sklearn.ensemble.GradientBoostingClassifier`

Construa um gráfico que apresenta a curva ROC para os três algoritmos acima. Use algum dos critérios de escolha de limiar apresentado em aula para definir o limiar de classificação para cada um desses algoritmos.

Após o treinamento e a escolha de limiares adequados, você deve avaliar a qualidade preditiva dos modelos correspondentes. Para isso, você deve usar os exemplos do conjunto de teste. 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 para cada modelo.
- Apresente também o relatório produzido pela função `classification_report` do Scikit-Learn.

2 Predição de preços de diamantes (2 pts)

Nessa parte, você deve treinar um modelo de regressão sobre o conjunto de dados Diamond. 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. (Por simplicidade, você pode manter os valores *default* dos hiperparâmetros de cada algoritmo. Lembre-se, entretanto)

1. `sklearn.linear_model.LinearRegression`
2. `sklearn.neighbors.KNeighborsRegressor`
3. `sklearn.ensemble.GradientBoostingRegressor`

Após o treinamento, você deve avaliar a qualidade preditiva de cada modelo de classificação resultante. Para isso, você deve previamente separar aleatoriamente 20% dos exemplos fornecidos para formarem o conjunto de teste. Isso permitirá que você obtenha uma estimativa do quão efetivos são os modelos gerados. Certifique-se de avaliar todos os modelos sobre o mesmo conjunto de teste.

1. Reporte o poder preditivo dos modelos que você construiu. Como métricas de avaliação, use o RMSE e o coeficiente e determinação R^2.
2. Apresente uma análise dos resultados obtidos. Como parte dessa análise, para cada um dos modelos gerados, construa gráficos conforme os exemplificados neste link.

3 Conjuntos desbalanceados - parte I (1,5 pts)

Nesta parte do trabalho, você deve usar o arquivo A652.pickle. Este arquivo 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. Nesse trecho de código, `filename` é o nome do arquivo fornecido.

```
import numpy as np
import pickle

f = open(filename, 'rb')

(X_train, y_train, X_val, y_val, X_test, y_test) = pickle.load(f)

print(f"Shapes: ", X_train.shape, X_val.shape, X_test.shape)
```

O atributo alvo corresponde a valores de precipitação (chuva). Você vai notar ao inspecionar as matrizes `y_*` de cada fonte que esses são conjuntos de dados para um problema de regressão (porque o alvo é um valor contínuo). Entretanto, nesta parte do trabalho, você deve alterar o alvo para gerar conjuntos de dados para classificação binária. Para isso, faça o seguinte, para cada arquivo fornecido. Se o alvo for igual a 0, altere para o rótulo **0**; em caso contrário, altere para o rótulo **1**. Após realizar essa transformação, você deve notar que os conjuntos de dados resultantes são extremamente desbalanceados.

Após realizar a transformação descrita acima, você deve investigar se a aplicação de alguma técnica de balanceamento de dados é efetiva no sentido de produzir um modelo de classificação 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.GradientBoostingClassifier`. Em sua análise dos resultados, 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)

Na parte 3, você enquadrou o problema de prever precipitação como uma tarefa de classificação binária. Nesta parte, você irá implementar um procedimento para criar um modelo de regressão para precipitação. Você deve usar o conjunto de dados fornecido no arquivo `A652.pickle`.

Você vai notar ao inspecionar as matrizes `y_train`, `y_val`, `y_test`, que elas apresentam valores contínuos não-negativos. Esses são valores de precipitação (chuva) medidos em mm/h. Você vai notar também que a grande maioria dos valores é igual a zero, o que corresponde

[1] A fonte desses dados e o modo pelo qual eles foram transformados para geração desses conjuntos são aspectos irrelevantes para o que deve ser feito nesta parte trabalho. Contudo, essas transformações serão alvo de estudo em aulas futuras do curso. Apenas para constar, esses dados correspondem a valores de variáveis meteorológicas (temperatura, pressão atmosférica, humidade relativa do ar, velocidade e direção do vento, precipitação) observados por uma estação de superfície do sistema INMET. O código identificador desta estação é A652.

a uma medição para a variável *precipitação* feita em um instante de tempo em que não está chovendo.

O procedimento para criar o modelo de regressão a ser usado nessa parte é descrito a seguir. Inicialmente você deve criar versões *binárias* das matrizes alvo `y_train`, `y_val`, `y_test`. Concretamente, para cada matriz alvo (`y_train`, `y_val`, `y_test`), substitua todos os valores diferentes de zero por **1** e os valores restantes por **0**. Ao fazer isso, você terá gerado matrizes resposta binárias. Chame essas matrizes de `y_train_bin`, `y_val_bin`, `y_test_bin`, respectivamente. Em seguida, execute os passos listados abaixo.

1. Treine um modelo de classificação binária usando (`X_train`, `y_train_bin`) como conjunto de treinamento. Chame esse modelo de $\mathcal{C}$.
2. Aplique $\mathcal{C}$ a todos os exemplos de `X_train`.
3. Defina `X_train_1` como o subconjunto de `X_train` que foi classificado por $\mathcal{C}$ como sendo da classe **1**. Defina também `y_train_1` como o subconjunto de `y_train` correspondente a `X_train_1`.
4. Treine um modelo de regressão usando (`X_train_1`, `y_train_1`) como conjunto de treinamento. Chame esse modelo de $\mathcal{R}$.
5. Para obter $\mathcal{R}'(\mathbf{x})$, i.e., a previsão de precipitação 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 os algoritmos que quiser para construir o classificador $\mathcal{C}$ e o regressor $\mathcal{R}$ mencionados no procedimento descrito acima. Reporte os resultados desse experimento para o conjunto de testes fornecido (`X_test`). Compare seu modelo de regressão produzido por meio desse procedimento com o modelo treinado usando (`X_train`, `y_train`) como conjunto de treinamento. Esse procedimento produziu um melhor modelo do ponto de vista preditivo? Como métrica de avaliação, use RMSE. Reporte valores dessa métrica para cada nível de severidade de precipitação, conforme a Tabela 2.

Tabela 2: Níveis de severidade de eventos de precipitação.

Precipitação (mm/h)	**Nível de Severidade**
$0 \leq x < 5$	Sem chuva / Leve
$5 \leq x < 25$	Moderada
$25 \leq x < 50$	Forte
$x \geq 50$	Tempestade

5 Calibração de modelos (1,5 pts)

Considere novamente o conjunto de dados fornecido na parte 1. No domínio do problema em questão, é importante que as probabilidades de predição do modelo estejam corretamente

calibradas. Sua tarefa nesta parte é produzir versões calibradas dos modelos originais criados na parte 1. Investigue o grau de calibração dos modelos originais e, conforme for o caso, aplique alguma técnica para calibrar os resultados do modelo. Você está livre para usar qualquer das técnicas de calibração que abordamos em aula, ou mesmo outra técnica que encontrar na literatura. Apresente gráficos para ilustrar os graus de calibração dos modelos antes e após aplicar a calibração. Apresente uma análise dos resultados obtidos.

6 Validação cruzada aninhada (1,0 pts)

Considere novamente o conjunto de dados Diamond apresentado na parte 2. Nessa parte do trabalho você deve realizar a validação cruzada aninhada (*nested cross-validation*) para encontrar (1) uma boa combinação de hiperparâmetros para ajustar um modelo e (2) um bom modelo para esse conjunto de dados. Como algoritmos de aprendizado candidatos, você deve escolher dois 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.