INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Serra

COORDENADORIA DE AUTOMAÇÃO INDUSTRIAL
Disciplina: Reconhecimento de Padrões
Período: 2024-2
Professor: Daniel Cruz Cavalieri

EXERCÍCIO 01: EDA, *Feature Engineering* e *Feature Selection*

OBS: recomenda-se que o trabalho seja feito em linguagem python e que os códigos sejam enviados juntamente com as respostas dos exercícios.

EDA:

As discussões de Aprendizado de Máquina geralmente são centradas em algoritmos e seu desempenho: como melhorar a precisão do modelo ou reduzir sua taxa de erro, excelência em engenharia de recursos ou ajuste fino de hiperparâmetros. Mas existe um conceito que vem antes de qualquer outra coisa: Análise Exploratória de Dados, ou EDA (do inglês *Exploratory Data Analysis*).

Este é um conceito central de Data Science, que às vezes é esquecido. O primeiro passo é conhecer seus dados: entendê-los, familiarizar-se com eles. Quais são as respostas que você está tentando obter com esses dados? Que variáveis você está usando e o que significam? Como isso parece de uma perspectiva estatística? Os dados estão formatados corretamente? Você tem valores ausentes? E duplicado? E quanto a outliers? Esse conceito fica ainda mais importante à medida que se aumenta o volume de dados.

Para esta parte do exercício iremos utilizar um conjunto de dados econômicos do Banco Mundial, descrevendo alguns fatores-chave mundiais, como PIB (em inglês GPD), níveis populacionais, superfície etc. O conjunto de dados se encontra no arquivo *wbdata.xlsx* e para a resolução do exercício algumas bibliotecas serão necessárias:

```
import pandas as pd
import numpy as np
import seaborn as sns
import xlrd
```

Passo 1: descreva os dados

Após carregar os dados, plote as 5 primeiras linhas da base de dados, extraia algumas estatísticas de cada coluna como: média; desvio padrão; mínimo e máximo. Além disso, verifique a forma (*shape*) da base dos dados e o tipo (*dtypes*) de cada coluna.

Passo 2: dados faltantes (https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4)

Verifique se há dados faltantes (nulos) ou NAN nas colunas da base de dados e elimine os objetos (linhas) da base de dados onde isso ocorreu. Verifique novamente o *shape* da base de dados.

Passo 3: visualização

Utilizando a função *pairplot* da biblioteca *Seaborn* plote um gráfico que relaciona cada características. Repare que a diagonal principal da figura gerada apresenta o histograma de cada característica. Faça uma análise das figuras e veja se existe alguma correlação entre os pares de características (gráfico linear).

Utilize agora o *boxplot* e verifique se existe *outliers* presentes nas características da base de dados. Aplique uma transformação logarítmica (*log transformation*) nos dados e faça novamente o *boxplot* das características. O que aconteceu com os *outliers*?

PCA:

As doenças cardíacas são uma das maiores causas de mortalidade entre a população do mundo. Isso faz das doenças cardíacas uma grande preocupação a ser tratada. Mas é difícil identificar doenças cardíacas devido a vários fatores de risco contributivos, como diabetes, pressão alta, colesterol alto, taxa de pulso anormal e muitos outros fatores. Devido a essas restrições, os cientistas se voltaram para abordagens modernas como *Data Mining* e *Machine Learning* para prever a doença.

O aprendizado de máquina mostra-se eficaz para auxiliar na tomada de decisões e previsões a partir da grande quantidade de dados produzidos pelo setor de saúde. Assim, neste exercício aplicaremos abordagens de *Machine Learning* (e eventualmente iremos compará-las) para classificar se uma pessoa está sofrendo de doença cardíaca ou não, usando um dos conjuntos de dados mais utilizados - o conjunto de dados *Cleveland Heart Disease* do *UCI Repository* (*cleveland.csv*).

1. Base de Dados

O conjunto de dados consiste em 303 dados individuais. Existem 14 colunas (13 colunas de atributos e 1 de classe), descritas a seguir:

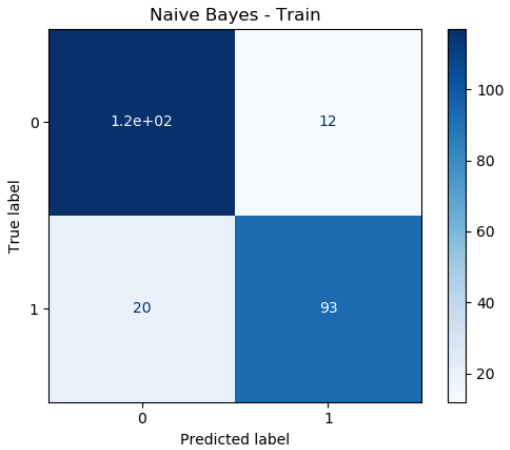
Coluna 1. Age
Coluna 2. Sex:
1 = male
0 = female
Coluna 3. Chest-pain type:
1 = typical angina
2 = atypical angina
3 = non — anginal pain
4 = asymptotic

Coluna 4. Resting Blood Pressure: value of an individual in mmHg (unit)
Coluna 5. Serum Cholestrol: displays the serum cholesterol in mg/dl (unit)
Coluna 6. Fasting Blood Sugar:
1 if blood sugar > 120mg/dl
0 otherwise
Coluna 7. Resting ECG:
0 = normal
1 = having ST-T wave abnormality
2 = left ventricular hyperthrophy
Coluna 8. Max heart rate achieved: max heart rate achieved by an individual.
Coluna 9. Exercise induced angina:
1 = yes
0 = no
Coluna 10. ST depression induced by exercise relative to rest: displays the value which is an integer or float.
Coluna 11. Peak exercise ST segment:
1 = upsloping
2 = flat
3 = downsloping
Coluna 12. Number of major vessels (0–3) colored by flourosopy: displays the value as integer or float.
Coluna 13. Thalassemia:
3 = normal
6 = fixed defect
7 = reversible defect
Coluna 14. **(Label)** Diagnosis of heart disease: Displays whether the individual is suffering from heart disease or not:
0 = absence
1, 2, 3, 4 = present.

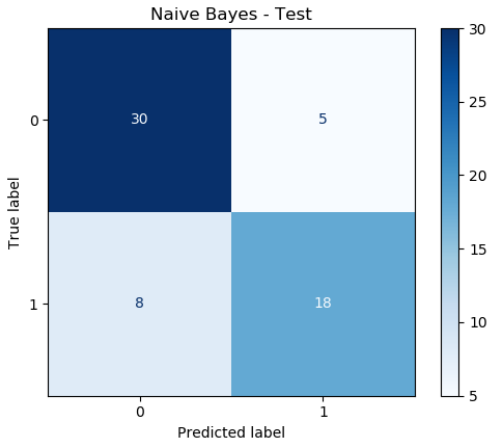
2. Código-exemplo

Utilizando o conjunto de dados c*leveland.csv* foi desenvolvido o código *código-exemplo-pca.py* para prever se uma pessoa terá ou não doença do coração. Neste código foi utilizado o algoritmo Naive Bayes sendo realizado um pré-processamento de dados (uma vez que existem dados faltantes). Com os dados carregados e corrigidos, eles foram separados em treinamento (80%) e teste (20%). O resultado do treinamento e teste são apresentados nas matrizes de confusão da tabela 1a e 1b, respectivamente.

Figura 1. Matriz de confusão geradas para o treinamento (a) e para o teste (b) do classificador Naive Bayes.



(a)



(b)

3. Pré-processamento dos dados

Adapte o código-exemplo para corrigir os dados faltantes. Para isso, utilize a função *fillna* da biblioteca *pandas* com a estratégia de média (*mean*). Especificamente, essa função irá substituir os dados faltantes pelo valor médio da coluna correspondente.

É importante lembrar que alguns algoritmos possuem melhores resultados com a normalização dos atributos. Assim, utilize a classe *StandardScaler* da biblioteca *scikit-learn* para a normalização dos dados. Com relação à normalização dos atributos responda:

- Qual o principal objetivo, dentro do contexto de distribuição, que esta função realiza?

4. Aplicação do PCA

Adapte o código-exemplo para inserir o algoritmo PCA. Utilize agora os dados transformados para treinar e testar novamente o algoritmo Naive Bayes. Apresente os resultados em formato de matriz de confusão e compare com os resultados obtidos anteriormente, sem a utilização do PCA.

Dados Desbalanceados:

Financial Distress é uma condição na qual uma empresa ou indivíduo não pode gerar receita porque é incapaz de cumprir ou não pode pagar suas obrigações financeiras. Isso geralmente se deve a altos custos fixos, ativos ilíquidos ou receitas sensíveis a crises econômicas. Ignorar os sinais de dificuldades financeiras pode ser devastador para uma empresa.

Pode chegar um momento em que dificuldades financeiras graves não possam ser sanadas porque as obrigações da empresa ou do indivíduo são muito altas e não podem ser pagas, e simplesmente não há receita suficiente para compensar a dívida. Se isso acontecer, a falência (ou *bankruptcy*) pode ser a única opção.

Este conjunto de dados lida com a previsão de dificuldades financeiras para uma amostra de empresas. Os atributos indicados por x1 a x83 são algumas características financeiras e não financeiras das empresas incluídas na amostra. Esses atributos pertencem ao período que deve ser usado para prever se a empresa terá problemas financeiros ou não. A primeira coluna é a variável categórica que representa a saída *healthy* (0) ou *bankruptcy* (1), que foi criada a partir da segunda coluna (*financial distress*).

Passo 1: carregar os dados

Carregar o arquivo *financial_distress.csv* e verificar se não há nenhum atributo com valor zero ou NAN. Caso haja dados faltantes, utilize uma das técnicas aprendidas na disciplina para substituir os dados. Além disso, é importante verificar que os atributos não estão normalizados, neste caso, faça uma normalização utilizando o z-*score* (média igual a 0 e desvio padrão igual a 1).

Passo 2: balanceamento das classes

Utilize os 2 tipos de balanceamento aprendidos na disciplina (*oversampling* e *undersampling*). Para verificar o efeito de cada técnica de balanceamento é necessário treinar um classificador. Neste caso, o passo 3 deverá ser seguido.

Passo 3: treinar um algoritmo de classificação

- A partir de cada técnica de balanceamento, separe os dados em treinamento (70%) e teste (30%). Treine um classificador kNN e obtenha a acurácia do teste para cada técnica de balanceamento. Discuta os resultados.
- Utilize agora a técnica de validação cruzada (*k-Fold Cross Validation*) com $k = 10$ e obtenha a acurácia média para cada técnica de balanceamento. Compare com os resultados obtidos anteriormente.
- Qual o problema da primeira estratégia utilizada (*Holdout*)?

Seleção de Atributos:

O ruído do aerofólio de aeronaves ocorre devido à interação entre uma lâmina do aerofólio e a turbulência produzida em sua própria camada limite. É o ruído total produzido quando um aerofólio encontra um fluxo suave e não turbulento.

Durante a última década, o Centro de Pesquisa Langley da NASA buscou desenvolver uma compreensão fundamental, bem como capacidade de previsão, dos vários mecanismos do auto-ruído (*self-noise airfoil*). O interesse foi motivado por sua importância para, por exemplo, rotores de helicóptero de banda larga, turbinas eólicas e o estudo de ruídos em estrutura.

O presente exercício é o resultado cumulativo de uma série de testes aerodinâmicos e acústicos de túnel de vento de seções de aerofólio, que produziram uma base de dados abrangente,

contendo 1503 exemplos com 5 atributos e uma saída, que é o nível de pressão sonora escalonada em decibéis.

Passo 1: carregar os dados

Carregar o arquivo *airfoil_self_noise.dat.* Serão carregados os seguintes dados de entrada:

Coluna 1: *Frequency*, em Hertzs.
Coluna 2: *Angle of attack*, em graus.
Coluna 3: *Chord length*, em metros.
Coluna 4: *Free-stream velocity*, em metro por segundos.
Coluna 5: *Suction side displacement thickness*, em metros.
Coluna 6 (saída): *Scaled sound pressure level*, em decibéis.

Passo 2: verificar a importância dos atributos

Utilizando o coeficiente de correlação de Pearson plote a importância de cada atributo para o problema dado. Discuta os resultados.

- Qual o problema de se utilizar o coeficiente de Pearson para a seleção de atributos?