Fundamentos de Data Science, Data Mining e Análise Preditiva

Especialização em Ciência de Dados com Big Data, BI e Data Analytics

Prof. Dr. Carlos Barros

Princípios data mining

- Introdução a Data Mining e Ciência dos Dados. Obtendo informações a partir dos dados. Principais Paradigmas e Modelos para mineração de dados. Dados incertos, com ruídos/outliers e confiança nos dados. Análise de dados exploratória. Introdução ao uso de modelos de predição. Escolha de modelos para mineração de dados. Redução de dimensionalidade e engenharia de dados. Minerando dados complexos. Trabalhando e limpando os Dados.

Redução de dimensionalidade e engenharia de dados. Minerando dados complexos. Trabalhando e limpando os Dados

Engenheiro de Dados

O Engenheiro de Dados é o responsável pela criação do pipeline que transforma os dados brutos que estão nos mais variados formatos, desde bancos de dados transacionais até arquivos de texto, em um formato que permita ao Cientista de Dados começar seu trabalho. Cabe também ao Engenheiro de Dados manter este pipeline em execução para que os dados possam ser coletados no momento certo, com o nível se segurança exigido pela empresa. O trabalho do Engenheiro de dados é tão importante quanto o trabalho do Cientista de Dados, mas eles costumam ter menos visibilidade, uma vez que estão mais distantes do produto final resultado do processo de análise, o que é produzido pelo Cientista de Dados.

Imagine uma aplicação e sua arquitetura de dados

- Entrada – isso envolve a coleta de dados necessários.
- Processamento – isso envolve o processamento dos dados para obter os resultados finais desejados.
- Armazenamento – isso envolve armazenar os resultados finais para recuperação rápida.
- Acesso – você precisará habilitar uma ferramenta ou usuário para acessar os resultados finais do pipeline.

Engenheiro de Dados

Habilidades do Engenheiro de Dados
Um Engenheiro de Dados precisa ser bom em:

- Arquitetar sistemas distribuídos
- Criar pipelines confiáveis
- Combinar fontes de dados
- Criar a arquitetura de soluções
- Colaborar com a equipe de Data Science e construir as soluções certas para essas equipes

Redução da dimensionalidade

Termo *dimensionalidade* é atribuído ao número de características de uma representação de padrões, ou seja, a dimensão do espaço de características. As duas principais razões para que a dimensionalidade seja a menor possível são: custo de medição e precisão do classificador. Quando o espaço de características contém somente as características mais salientes, o classificador será mais rápido e ocupará menos memória.

Redução da dimensionalidade

Com o aumento horizontal das bases de dados (dimensões / atributos) um problema grave é o aumento da dimensionalidade (Course of Dimensionality) em que temos não somente multicolinearidade, heteroscedasticidade e autocorreação para ficar em exemplos estatísticos simples. Em termos computacionais nem é preciso dizer que o aumento de atributos faz com que os algoritmos de Data Mining ou Inteligência Computacional tenham que processar um volume de dados muito maior (aumento da complexidade do processamento = maior custo temporal).

Redução da dimensionalidade – Principais técnicas

Missing Values Ratio.
Low Variance Filter
High Correlation Filter
Random Forests / Ensemble Trees
Principal Component Analysis (PCA)
Backward Feature Elimination.
Forward Feature Construction

Redução da dimensionalidade

knime_seventechniquesdatadimreduction

Dimensionality Reduction	Reduction Rate	Accuracy on validation set	Best Threshold	AuC	Notes
Baseline	0%	73%	-	81%	Baseline models are using all input features
Missing Values Ratio	71%	76%	0.4	82%	-
Low Variance Filter	73%	82%	0.03	82%	Only for numerical columns
High Correlation Filter	74%	79%	0.2	82%	No correlation available between numerical and nominal columns
PCA	62%	74%	-	72%	Only for numerical columns
Random Forest / Ensemble Trees	86%	76%	-	82%	-
Backward Feature Elimination + missing values ratio	99%	94%	-	78%	Backward Feature Elimination and Forward Feature Construction are **prohibitively slow** on high dimensional data sets. It becomes practical to use them, only if following other dimensionality reduction techniques, like here the one based on the number of missing values.
Forward Feature Construction + missing values ratio	91%	83%	-	63%	

Apesar da robustez matemática, o PCA apresenta um resultado não tão satisfatório em relação a métodos mais simples de seleção de atributos. Isso pode indicar que esse método não lida tão bem com bases de dados com inconsistências. Filtro de baixa variância e de valores faltantes são técnicas absolutamente simples e tiveram o mesmo resultado de técnicas algoritmicamente mais complexas como Florestas Aleatórias. Construção de modelos com inclusão incremental de atributos e eliminação de atributos retroativa são métodos que apresentam uma menor performance e são proibitivos em termos de processamento.

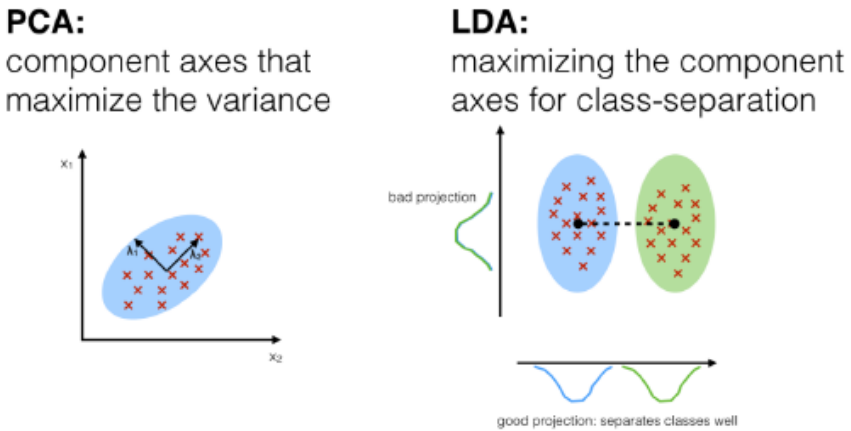
A estatística básica ainda é uma grande ferramenta para qualquer data miner, e não somente ajuda em termos de redução do custo temporal (processamento) quanto em custo espacial (custo de armazenamento).

PCA (Principal Component Analysis)

- Seleção de características x Extração de características
- Um dos principais algoritmos de aprendizagem de máquina não supervisionada
- Identifica a correlação entre variáveis, e caso haja uma forte correlação é possível reduzir a dimensionalidade
- Das ***m*** variáveis independentes, PCA extrai ***p*** <= ***m*** novas variáveis independentes que explica melhor a variação na base de dados, sem considerar a variável dependente
- O usuário pode escolher o número de ***p***

LDA (Linear Discriminant Analysis)

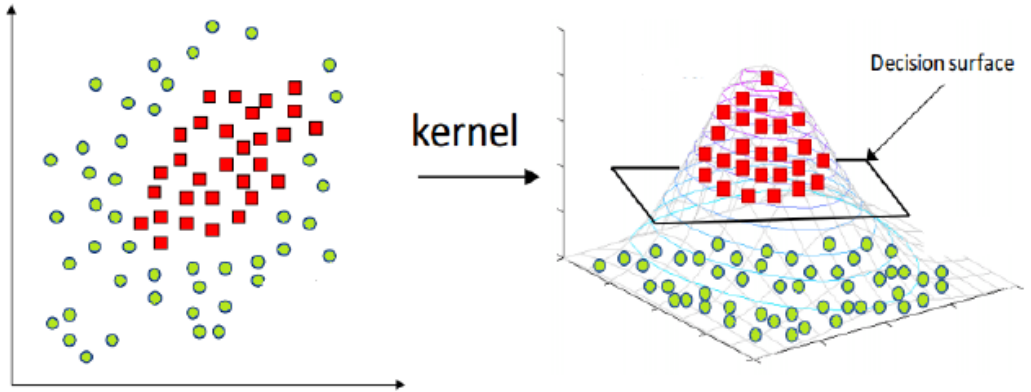
- Além de encontrar os componentes principais, LDA também encontra os eixos que maximizam a separação entre múltiplas classes
- É supervisionado por causa da relação com a classe
- Das ***m*** variáveis independentes, LDA extrai ***p <= m*** novas variáveis independentes que mais separam as classes da variável dependente



Fonte: https://sebastianraschka.com/Articles/2014_python_lda.html

Kernel PCA

- PCA e LDA são utilizados quando os dados são linearmente separáveis
- Kernel PCA é uma versão do PCA que os dados são mapeados para uma dimensão maior usando o ***kernel trick***
- Os componentes principais são extraídos dos dados com dimensionalidade maior



```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.svm import SVC
from sklearn.ensemble import ExtraTreesClassifier

credito = pd.read_csv('Credit.csv')
previsores = credito.iloc[:,0:20].values
classe = credito.iloc[:,20].values

labelencoder = LabelEncoder()
previsores[:,0] = labelencoder.fit_transform(previsores[:,0])
previsores[:,2] = labelencoder.fit_transform(previsores[:,2])
previsores[:, 3] = labelencoder.fit_transform(previsores[:, 3])
previsores[:, 5] = labelencoder.fit_transform(previsores[:, 5])
previsores[:, 6] = labelencoder.fit_transform(previsores[:, 6])
previsores[:, 8] = labelencoder.fit_transform(previsores[:, 8])
previsores[:, 9] = labelencoder.fit_transform(previsores[:, 9])
previsores[:, 11] = labelencoder.fit_transform(previsores[:, 11])
previsores[:, 13] = labelencoder.fit_transform(previsores[:, 13])
previsores[:, 14] = labelencoder.fit_transform(previsores[:, 14])
previsores[:, 16] = labelencoder.fit_transform(previsores[:, 16])
previsores[:, 18] = labelencoder.fit_transform(previsores[:, 18])
previsores[:, 19] = labelencoder.fit_transform(previsores[:, 19])

X_treinamento, X_teste, y_treinamento, y_teste = train_test_split(previsores,
                                                                  classe,
                                                                  test_size = 0.3,
                                                                  random_state = 0)
svm = SVC()
svm.fit(X_treinamento, y_treinamento)
previsoes = svm.predict(X_teste)
taxa_acerto = accuracy_score(y_teste, previsoes)

forest = ExtraTreesClassifier()
forest.fit(X_treinamento, y_treinamento)
importancias = forest.feature_importances_

X_treinamento2 = X_treinamento[:,[0,1,2,3]]
X_teste2 = X_teste[:,[0,1,2,3]]

svm2 = SVC()
svm2.fit(X_treinamento2, y_treinamento)
previsoes2 = svm2.predict(X_teste2)
taxa_acerto = accuracy_score(y_teste, previsoes2)
```

Dados complexos – dados estr & não estruturados (80%)

dados multimídia (imagens, video, audio) – netlfix, clinicas, tv, radios
dados geográficos -
dados temporais - funceme
Dados do genoma -
Textos

Algumas aplicações onde o controle e acesso a informações temporais são fundamentais:

- **Área Médica**
- **Área Empresarial**
  - Aplicações Financeiras
  - Controle de Produção
  - Gerenciamento de Vendas
  - Gestão de Pessoas
- **Controle Acadêmico**
- **Sistemas de Informações Geográficas**
- **Sistema de Reservas**
- (...)

O LHC (Large Hadron Collider) é um acelerador de partículas instalado próximo da fronteira entre Suíça e França. Ele contém quatro detetores de partículas que registram 40 milhões de eventos por segundo, registrados por 150 milhões de sensores. O volume de dados pré-processados é aproximadamente igual a 27 terabytes por dia.

O Instituto Nacional de Pesquisas Espaciais tem uma base de dados de imagens de satélite com mais de 130 terabytes [29].

O projeto Internet Archive3 mantém um arquivo de diversos tipos de mídia, contendo 2 petabytes e crescendo cerca de 20 terabytes por mês, com aproximadamente 130.000 vídeos, 330.000 arquivos de áudio, quase 500.000 documentos de texto e indexando 85 bilhões de páginas em várias versões.

De acordo com algumas estimativas4, o site YouTube continha 45 terabytes de vídeos em 2006. O site Flickr tinha 2 bilhões de fotografias digitais5 em 2007 (e um teste rápido mostrou que já são ao menos 2.2 bilhões). Considerando que uma imagem, suas variantes criadas pelo site e outros dados como comentários ocupem um mínimo de 300 kilobytes, toda a coleção usa mais de 614 terabytes no total.

O banco de dados GenBank contém coleções anotadas de sequências de nucleotídeos e proteínas de mais de 100.000 organismos, em um total de 360 gigabytes.

O Large Synoptic Survey Telescope contém uma câmera digital de aproximadamente 3.2 gigapixels e deve coletar 20 a 30 terabytes de imagens por noite.

Um levantamento feito pelaWinter Corporation9 menciona algumas bases de dados de grande porte em uso (em 2005): Yahoo! (100 terabytes), AT&T (93 terabytes), Amazon (24 terabytes), Cingular (25 terabytes).

Minenação de textos

Análise de sentimentos
Classificação de documentos
detecção de fraudes
Anúncios contextualizados
Filtro de Spam

```
import matplotlib.pyplot as plt
import nltk
#nltk.download()
from nltk.corpus import PlaintextCorpusReader
from nltk.corpus import stopwords
from matplotlib.colors import ListedColormap
from wordcloud import WordCloud
import string

corpus = PlaintextCorpusReader('Arquivos', '.*')

arquivos = corpus.fileids()
arquivos[0]
arquivos[0:100]
for a in arquivos:
    print(a)

texto = corpus.raw('1.txt')

todo_texto = corpus.raw()
palavras = corpus.words()
palavras[170]
len(palavras)

stops = stopwords.words('english')
mapa_cores = ListedColormap(['orange', 'green', 'red', 'magenta'])
nuvem = WordCloud(background_color = 'white',
                  colormap = mapa_cores,
                  stopwords = stops,
                  max_words = 100)
nuvem.generate(todo_texto)
plt.imshow(nuvem)

palavras_semstop = [p for p in palavras if p not in stops]
len(palavras_semstop)

palavras_sem_pontuacao = [p for p in palavras_semstop if p not in string.punctuation]
len(palavras_sem_pontuacao)

frequencia = nltk.FreqDist(palavras_sem_pontuacao)
mais_comuns = frequencia.most_common(100)
```

Tarefa 3

Leia os dados do arquivo data1.csv A classe de cada dado é o valor da última coluna (0 ou 1).
1. faca o PCA dos dados (sem a última coluna). Se você quiser que os dados transformados tenham 80% da variância original, quantas dimensões do PCA você precisa manter?
Gere os dados transformados mantendo 80% da variância. (Atenção este passo não é 100% correto do ponto de vista de aprendizado de maquina. Não repita este passo em outras atividades).
Considere as primeiras 200 linhas dos dados como o conjunto de treino, e as 276 ultimas como o conjunto de dados
2. Treine uma regressão logística no conjunto de treino dos dados originais e nos dados transformados. Qual a taxa de acerto no conjunto de teste nas 2 condições (sem e com PCA)?
3. Treine o LDA, ExtraTreesClassifier e F-Test nos conjuntos de treino com e sem PCA e teste nos respectivos conjuntos de testes. Qual a acurácia nas 2 condições?
4. Qual a melhor combinação de classificador e PCA ou não?