Aprendizagem Automática

Trabalho Laboratorial – grupos de 1 ou 2 alunos

Classificação de Críticas de Cinema do IMDb

1º Semestre de 2020/2021

1 Objectivos do trabalho

Este trabalho lida com textos de críticas de cinema do IMDb, e está dividido em duas tarefas de classificação. Baseado nos textos dos documentos, pretende-se:

I. Determinar se a crítica é positiva ou negativa (classificação binária).

II. Prever a pontuação da crítica (classificação multi-classe). Para tal considere que existem 8 classes, compostas pelas críticas com pontuações de 1-4 e de 7-10.

2 Dados

A IMDb, a Internet Movie Database, é uma base de dados que consiste em textos de críticas de cinema, recolhidas por Andrew Mass [1]. Neste trabalho, os dados são compostos por 40 000 textos de críticas de cinema com as respetivas pontuações, e encontram-se no ficheiro `imdbCriticas.p`. Este ficheiro contém uma variável do tipo dicionário com dois campos: `data` lista com os textos de criticas de cinema, `target` com a pontuação da crítica. Para a tarefa de classificação binária, considere que críticas positivas têm uma pontuação superior ou igual a 7 (numa escala de 1 a 10) e críticas negativas uma pontuação inferior ou igual a 4. Críticas neutras (pontuações 5 e 6) foram excluídas.
Para mais informação, consultar: `ai.stanford.edu/~amaas/data/sentiment/`

3 Desenvolvimento

Deverá ter em conta os seguintes pontos:

1. Construção do vocabulário:
   (a) Modifique o valor dos parâmetros `min_df` e `token_pattern` na função `TfidfVectorizer` para obter vocabulários de diferentes dimensões. Analise como o desempenho dos modelos projetados é afetado pela dimensão do vocabulário. Deve igualmente averiguar qual a dimensão mínima do vocabulário para

a qual o desempenho dos classificadores binários seja ainda próximo dos melhores resultados obtidos.

(b) Teste se a inclusão de n-gramas é benéfico para o desempenho dos modelos projetados.

2. Metodologias de teste e métricas de desempenho:
   (a) Escolher a metodologia de teste apropriada de modo a ter uma estimação fidedigna do desempenho dos modelos treinados.
   (b) No problema de classificação binária, usar as métricas apropriadas e calibrar os modelos treinados.

3. Classificadores:
   (a) Obrigatório: testar um discriminante logísticos nas duas tarefas de classificação.
   (b) Grupos individuais: testar mais um classificador.
   (c) Grupos de dois alunos: testar mais dois classificadores.

4 Pontuação

A pontuação pode ser alterada mediante a discussão do projeto.

REQUISITOS MÍNIMOS:

O código deverá estar num ficheiro `.ipynb` (Jupyter Notebook).

10 valores

- Implementação e avaliação dos classificadores para as tarefas de classificação binária e multi-classe. Necessário o bom funcionamento de todos os programas e o cumprimento dos seguintes pontos:
  - Programa(s) de conversão de uma *string* de texto (ou uma listas de *strings*) na representação tf-idf.
  - Programa(s) de treino e de avaliação dos classificadores nas tarefas de classificação multi-classe e binária.
  - O(s) programa(s) de avaliação devem expor claramente os resultados obtidos, preferencialmente através de gráficos ou imagens.
- Apresentação (slides/PowerPoint) com a descrição das experiências efetuadas e dos resultados obtidos.
  - Grupos de 1: Apresentação com o máximo de 20 slides.
  - Grupos de 2: Apresentação com o máximo de 30 slides.

A apresentação deve ser entregue num ficheiro .pdf com o nome: Axxxxx.pdf (ou AxxxxxAxxxxx.pdf - para grupos de 2 alunos). Tenha em conta a estrutura da apresentação: devem estar claramente identificadas as tarefas abordadas, descritas as experiência efetuadas, métodos usados, resultados obtidos, etc.

VALORES ADICIONAIS:

+ 2 valor Jupyter Notebook: clareza da apresentação, dos comentários e do código.

+ 4 valores Grupos individuais: escolher 1 tópico.
Grupos de 2 alunos: escolher 2 tópicos.

REGRESSÃO Considere que a tarefa de estimar a pontuação da crítica é um problema de regressão. Treine e avalie um modelo de regressão linear. Repita o processo com um modelo de regressão não linear à sua escolha. Compare os resultados da regressão com os obtidos no problema de classificação multi-classe.

PCA Investigue se o pré-processamento dos dados com PCA, é benéfico para o desempenho de discriminantes logísticos na tarefa de classificação binária e multi-classe. Determine igualmente qual o número ótimo de componentes principais. Nota: use a função `TruncatedSVD` em vez de `PCA` do submódulo `sklearn.decomposition` para poder lidar com matrizes esparsas.

DIMENSÃO do VOCABULÁRIO O objetivo deste tópico é investigar a influência do tamanho do dicionário (dimensão dos dados) no desempenho de um discriminante logístico no problema de classificação binária. Comece por construir um dicionário com dimensão elevada ($>>$ 30 000 dimensões) - use *stemming* e inclua n-gramas na representação tf-idf dos textos. Use regularização *lasso* para descartar dimensões não discriminativas e consequentemente reduzir a dimensão do dicionário. Averigue até onde se pode reduzir a dimensão do dicionário sem que isso resulte numa degradação do desempenho significativa. Compare os resultados da classificação binária, obtidos com os dicionário de diferentes dimensões recorrendo à regularização *lasso*, com outros dicionários com as mesmas dimensões mas obtidos definindo o parâmetro `max_features` da função `TfidfVectorizer`.

CLUSTERING Use um ou mais algoritmos de *clustering* à sua escolha para agrupar críticas de uma forma não supervisionada. Analise os resultados e indique os grupos em

que as críticas abordam um tópico específico. Investigue o efeito da variação do número de *clusters* no desempenho dos algoritmos de agrupamento.

+ 4 valores CONHECIMENTOS ADQUIRIDOS:

Conhecer, saber explicar e como aplicar os seguintes tópicos/métodos no contexto das críticas IMDb.

- Métodos de pré-processamento de dados.
- Métodos de aprendizagem supervisionada.
- Métodos de aprendizagem não supervisionada.
- Calibração e comparação de modelos de classificação binária.
- Metodologias de treino e teste.
- Análise dos resultados obtidos.

+2 extra COMPETIÇÃO:

- Os dez grupos que obtiverem a melhor probabilidade de acertos na tarefa de classificação binária terão mais 1 valor na nota final.
- Os dez grupos que obtiverem a melhor probabilidade de acertos na tarefa de classificação multi-classe terão mais 1 valor na nota final.
- Podem usar todos os dados disponibilizados (ficheiro `imbdCriticas.p`) para o treino dos classificadores (ver próximas funções). Não é permitidos usar outros dados. O uso de outros dados resulta na exclusão da competição.

CÓDIGO:

O código em Python deverá ser entregue num Jupyter Notebook.

-1 valor Adicionalmente, devem ser estar definidas as seguintes funções que serão usadas na competição. O mau funcionamento resulta num desconto de 1 valor na nota final (-1 por função).

- `X=text2vector(Docs)`

  Função que converte uma lista de *strings* (variável `Docs`) na representação tf-idf. Deve retornar uma matriz documento/termo (matriz `X`). A função deverá fazer as limpezas necessárias aos documentos (*strings*) e usar um modelo tf-idf **previamente treinado**. A matriz será depois usada nas funções de classificação binária e multi-classe. Caso seja necessário executar o treino do modelo tf-idf antes de chamar a função, esta será considerada como não estando a funcionar.

- `y=binClassify(X)`
  Função que classifica a matriz `X` da função anterior em positivos e negativos. Deve retornar um *array*, `y`, de 0s e 1s com o resultado da classificação. Pode usar um classificador que achar adequado, **mas este já deve estar treinado**. Caso seja necessário executar o treino antes de chamar a função, esta será considerada como não estando a funcionar.
- `y=multiClassify(X)`
  Função que classifica a matriz `X` devolvida pela função (`text2vector`) numa de oito classes (1-4 e 7-10). Deve retornar um *array*, `y`, de inteiros (1-4 e de 7-10) com o resultado da classificação. Pode usar um classificador que achar adequado, **mas este já deve estar treinado**. Caso seja necessário executar o treino antes de chamar a função, esta será considerada como não estando a funcionar.

BIBLIOTECAS:

Bibliotecas de Python permitidas: `numpy`, `scipy`, `matplotlib`, `sklearn`, `nltk`, `re` e `opencv`.

-1 valor Casos seja necessário instalar outras bibliotecas, haverá uma penalização de 1 valor.

5 Ficheiros a Entregar e Outros Pontos

-1 valor
- Deve ser entregue, via Moodle, um único ficheiro zip denominado `AxxxxxProjeto.zip` (ou `AxxxxxAxxxxxProjeto.zip` para grupos de 2 alunos).
- O ficheiro zip deverá conter os seguintes ficheiros:
  -1 valor – Apresentação: `Axxxxx.pdf` ficheiro pdf com slides de apresentação.
  -1 valor – Jupyter Notebook: `Axxxxx.ipynb` (ou `AxxxxxAxxxxx.ipynb` para grupos de 2 alunos).
- Pode (deve) igualmente incluir no zip outros ficheiros, como modelos pré-treinados e de dados. Tenha em atenção para que o tamanho do ficheiro zip não exceda o limite de 100MB.

Referências

[1] Maas, Andrew L. and Daly, Raymond E. and Pham, Peter T. and Huang, Dan and Ng, Andrew Y. and Potts, Christopher, *Learning Word Vectors for Sentiment Analysis*, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011.