rna-cdroput

Técnicas de Aprendizado de Máquina para Predição Desvio Fotométrico

Introdução • Funcionalidades • Como usar • Exemplo • Créditos

Introdução

O rna-cdroput é um repositório criado para implementação da dissertação de mestrado de Rafael Fialho (@rapguit). O trabalho **Técnicas de Aprendizado de Máquina para Predição Desvio Fotométrico** acrescenta o conceito e a implementação de diferentes estratégias para predição de erros para valores de magnitudes.

Funcionalidades

- Download e conversão de conjunto de dados
  - SDSS
  - Happy
  - Teddy
- Predição de erros para valores de magnitudes
  - Implementação de três estratégias
  - Implementação de cinco métodos de regressão para cada estratégia
- Divisão do conjunto de dados em conjuntos de teste e treino
  - Script configurado para 80% treino e 20% teste
- Transformação dos valores para otimizar treinamento

- Treinamento de modelos para predição de desvio fotométrico
- Selação de modelos
- Predição de desvios fotométricos

Como usar

Para clonar e rodar essa aplicação, você irá precisar do git e asdf. Pelo seu terminal:

```
# Clone o repositório
git clone https://github.com/MLRG-CEFET-RJ/rna-cdroput

# Entre no repositório
cd rna-cdroput

# Instale o plugin do python no asdf

asdf plugin-add python

# Instale a versão do python utilizada no projeto

asdf install python 3.10.5

# Faça um ambiente virtal python isolado

python -m venv venv

# Ative o ambiente virtual isolado

source venv/bin/activate

# Instale as dependências

pip install -r requirements.txt

# Faça o download dos conjuntos de dados
source src/scripts/download.sh

# Rode o seguinte script shell para experimentos rápidos

source src/scripts/predict_fast.sh
```

Exemplo

Nessa seção são apresentados dois exemplos, um simples e outro o próprio script shell utilzado nos experimentos.

Exemplo 1 (script de uma linha simples)

```
python -m src.strategies.ir_1_x_1 -dataset teddy_data.csv
```

Exemplo 2 (script utilizado nos experimentos)

```
declare -a REGRESSORS=("dt" "knn" "mlp" "rf" "xgb")
declare -a STRATEGIES=("1_x_1" "m_x_1" "m_x_m")
declare -a DATASETS=("teddy" "happy")

for dataset in "${DATASETS[@]}"
do
    for regressor in "${REGRESSORS[@]}"
    do
        for strategy in "${STRATEGIES[@]}"
        do
            eval "python -m src.strategies.${regressor}_${strategy} -dataset ${dataset
            eval "python -m src.strategies.${regressor}_${strategy} -dataset ${dataset
            eval "python -m src.strategies.${regressor}_${strategy} -dataset ${dataset
            eval "python -m src.strategies.${regressor}_${strategy} -dataset ${dataset

            eval "python -m src.pipeline.dataset_split -dataset ${dataset}_data_${regr

            eval "python -m src.pipeline.dataset_scaling -datafiles ${dataset}_data_${

            eval "python -m src.training -n ${dataset}_training_${regressor}_${strateg

            BEST_MODELS=$(python -m src.modules.best_model_selector -n ${dataset}_trai

            eval "python -m src.predict -n ${dataset}_B -models ${BEST_MODELS} -testse
            eval "python -m src.predict -n ${dataset}_C -models ${BEST_MODELS} -testse
            eval "python -m src.predict -n ${dataset}_D -models ${BEST_MODELS} -testse
        done
    done
done

### ISOTONIC REGRESSOR ###
# ISOTONIC REGRESSOR DOES NOT SUPPORT MULTIPLE INPUT/OUTPUT!

### TEDDY ###
python -m src.strategies.ir_1_x_1 -dataset teddy_data.csv
python -m src.strategies.ir_1_x_1 -dataset teddy_test_data_B.csv
python -m src.strategies.ir_1_x_1 -dataset teddy_test_data_C.csv
python -m src.strategies.ir_1_x_1 -dataset teddy_test_data_D.csv

python -m src.pipeline.dataset_split -dataset teddy_data_ir_1_x_1_experrs.csv -p 80:20

python -m src.pipeline.dataset_scaling -datafiles teddy_data_ir_1_x_1_experrs_train.cs

python -m src.training -n teddy_training_ir_1_x_1 -e 5000 -dp ErrorBasedInvertedRandom

BEST_MODELS=$(python -m src.modules.best_model_selector -n teddy_training_ir_1_x_1 -e
```

```
eval "python -m src.predict -n teddy_B -models ${BEST_MODELS} -testset teddy_test_data
eval "python -m src.predict -n teddy_C -models ${BEST_MODELS} -testset teddy_test_data
eval "python -m src.predict -n teddy_D -models ${BEST_MODELS} -testset teddy_test_data

### HAPPY ###
python -m src.strategies.ir_1_x_1 -dataset happy_data.csv
python -m src.strategies.ir_1_x_1 -dataset happy_test_data_B.csv
python -m src.strategies.ir_1_x_1 -dataset happy_test_data_C.csv
python -m src.strategies.ir_1_x_1 -dataset happy_test_data_D.csv

python -m src.pipeline.dataset_split -dataset happy_data_ir_1_x_1_experrs.csv -p 80:20

python -m src.pipeline.dataset_scaling -datafiles happy_data_ir_1_x_1_experrs_train.cs

python -m src.training -n happy_training_ir_1_x_1 -e 5000 -dp ErrorBasedInvertedRandom

BEST_MODELS=$(python -m src.modules.best_model_selector -n happy_training_ir_1_x_1 -e

eval "python -m src.predict -n happy_B -models ${BEST_MODELS} -testset happy_test_data
eval "python -m src.predict -n happy_C -models ${BEST_MODELS} -testset happy_test_data
eval "python -m src.predict -n happy_D -models ${BEST_MODELS} -testset happy_test_data

### GENERATE ERROR RESULTS ###
python -m src.report.error_results
```

Créditos

Essa aplicação utiliza os seguintes projetos de código aberto:

- Git
- asdf
- Python

E os seguintes pacotes, módulos e bibliotecas de código aberto para Python:

```
absl-py==1.1.0
astunparse==1.6.3
beautifulsoup4==4.11.1
cachetools==5.2.0
certifi==2022.6.15
charset-normalizer==2.1.0
cycler==0.11.0
filelock==3.7.1
flatbuffers==1.12
fonttools==4.34.4
gast==0.4.0
gdown==4.5.1
google-auth==2.9.0
```

```
google-auth-oauthlib==0.4.6
google-pasta==0.2.0
grpcio==1.47.0
h5py==3.7.0
idna==3.3
Jinja2==3.1.2
joblib==1.1.0
keras==2.9.0
Keras-Preprocessing==1.1.2
kiwisolver==1.4.3
libclang==14.0.1
Markdown==3.3.7
MarkupSafe==2.1.1
matplotlib==3.5.2
numpy==1.23.1
oauthlib==3.2.0
opt-einsum==3.3.0
packaging==21.3
pandas==1.4.3
Pillow==9.2.0
protobuf==3.19.4
pyasn1==0.4.8
pyasn1-modules==0.2.8
pyparsing==3.0.9
PySocks==1.7.1
python-dateutil==2.8.2
pytz==2022.1
requests==2.28.1
requests-oauthlib==1.3.1
rsa==4.8
scikit-learn==1.1.1
scipy==1.8.1
seaborn==0.11.2
six==1.16.0
sklearn-pandas==2.2.0
soupsieve==2.3.2.post1
tensorboard==2.9.1
tensorboard-data-server==0.6.1
tensorboard-plugin-wit==1.8.1
tensorflow==2.9.1
tensorflow-estimator==2.9.0
tensorflow-io-gcs-filesystem==0.26.0
termcolor==1.1.0
threadpoolctl==3.1.0
tqdm==4.64.0
typing_extensions==4.3.0
urllib3==1.26.10
Werkzeug==2.1.2
wrapt==1.14.1
xgboost==1.6.1
```

> Cássio Souza @cassiofb-dev · Jorge Gabriel @jorge-g99 · Rafael Fialho @rapguit