README

Name : Tushar Kumar
NetId : tusharku
Collaborators : None

December 13, 2018

1 Introduction

This project, which was implemented by myself, and **did not have any collaboration from anybody else**, implements seven machine learning algorithms and analyzes the result of those algorithms on UCI online News Popularity dataset. I also analyze the dataset and mine for properties and relevance information with respect to the prediction task of determining if an article is popular or not based on its attributes. The algorithms that I implement are Decision Trees, Logistic Regression, Neural Net, Neural Nets with glove embeddings of url as features, Random Forest and Extra Trees classifier. This project was undertaken as part of the graduate course(CSC 440) at University of Rochester.

2 Technology Used

- Python as programming language(3.7.0)
- PyCharm used as IDE
- Overleaf for generating reports in LaTeX

3 Project dependencies

To install all requirements other than Python3 and skfeature , you can use the requirements.txt file in the folder provided by using the below mentioned command :

```
cd CSC440FinalProject-tusharku

pip3 install -r requirements.txt
```

For skfeature, please kindly use the link mentioned below

- Python 3.7.0

- numpy 1.15.1
- Pytorch 0.4.1 - Get Pytorch
- scipy - 1.1.0 Get Scipy
- sklearn 0.20.0 - Get Sklearn
- skfeature - Get Skfeature
- Matplotlib 2.2.3 - Get Matplotlib

4 Running the project

- Download the **CSC440FinalProject-tusharku-tusharku.zip** file(which you would have if you are reading this file.
- Unzip the file to get the CSC440FinalProject-tusharku folder
- Run the below mentioned command - Please mind the line break created because the command being of greater length than width of the page.

```
cd CSC440FinalProject-tusharku

python3 -m com.uofr.course.csc440.project.newspopularity.runModels
↪  --train_test_split 0.2 --model dtree --max_depth_tree 5
```

This, would run the decision tree algorithm on the News popularity dataset with a split of 80/20 for training and testing.

5 Executing Models

I would now provide a detailed description of all the command line arguments and how to use them to evaluate the different project requirements and run the different models These are the commandline arguments with their description and possible values that can be provided in order to run the models.

5.1 Command line arguments

- **train_test_split** : What is the split you want to use to train the model and test.
  Possible Values : float (between 0 and 1)
  Default Value : 0.2(so a 80/20 split)

- **model** : Which model you want to use for learning
  Possible Values : dtree(for decision tree) or logistic(for logistic regression) or nnet(for neural net) or forest(for random forest) or etree(for extra tree classifier)
  Default Value : dtree

- **log_interval** : After how many epochs do you wish to log the training loss and testing accuracy for algorithms which learn in epochs
  Possible Value : int
  Default Value : 10

- –**batch_size** : What batch size you want to use
  Possible Value : int
  Default Value : 100

- **epochs** : Number of epochs you want to run the learning process for
  Possible Values : Integer
  Default Value : 150

- **lr** : Learning rate you want to use
  Possible Values : float
  Default Value : 0.001

- **seed** : Seed value you wish to use
  Possible Values : int
  Default Value : 0

- **max_depth** : Maximum depth the decision tree should be allowed to grow to
  Possible Values : int
  Default Value : 5

- **max_depth_forest** : Maximum depth the trees in random forest should be allowed to grow to
  Possible Values : int
  Default Value : 26

- **max_depth_extra_tree** : Maximum depth the trees in extra tree classifier forest should be allowed to grow to
  Possible Values : int
  Default Value : 22

- **trees_forest** : Number of weak learners to use for Random Forest classifier
  Possible Values : int

Default Value : 550

- **trees_extra** : Number of weak learners to use for Extra Tree classifier
  Possible Values : int
  Default Value : 500

- **dropout** : Dropout to be used for neural net
  Possible Values : float
  Default Value : 0

- **weight_decay** : Weight decay to be used
  Possible Values : float
  Default Value : 0

- **k** : Top k(based on fisher score) features will be chosen for training
  Possible Values : int
  Default Value : 61(all features)

- **use_embedding** : If present in the command the program will use the url embeddings using glove vectors for neural network
  Default Value : False. URL Glove embeddings will not be used

5.1.1 Running Models

1. **Decision Trees with optimized hyperparameters** :

```
python3 -m com.uofr.course.csc440.project.newspopularity.runModels
 ↪  --train_test_split 0.2 --model dtree --max_depth_tree 5 --seed 0
```

2. **Logistic Regression with all attributes with optimized hyperparameters**

```
python3 -m com.uofr.course.csc440.project.newspopularity.runModels
 ↪  --train_test_split 0.2 --model logistic --log_interval 10
 ↪  --batch_size 100 --epochs 250 --lr 0.001 --seed 0 --dropout 0
 ↪  --weight_decay 0.0001
```

3. **Logistic Regression with top 36 attributes with optimized hyperparameters**

```
python3 -m com.uofr.course.csc440.project.newspopularity.runModels
 ↪  --train_test_split 0.2 --model logistic --log_interval 10
 ↪  --batch_size 100 --epochs 250 --lr 0.001 --seed 0 --dropout 0
 ↪  --weight_decay 0.0001 --k 36
```

4. **Neural Network with optimized hyperparameters**

```
python3 -m com.uofr.course.csc440.project.newspopularity.runModels
 ↪  --train_test_split 0.2 --model nnet --log_interval 10 --batch_size
 ↪  100 --epochs 150 --lr 0.001 --seed 0 --dropout 0.2 --weight_decay
 ↪  0.0001
```

5. **Neural Network with url glove embeddings and optimized hyperparameters** For using glove embeddings, you must download the glove vectors from the url Glove Vectors. Once downloaded please move it to the folder CSC440FinalProject-tusharku and unzip it and then move the 25d.txt file by running the following command

```
cd CSC440FinalProject-tusharku
 mv glove.twitter.27B/glove.twitter.27B.25d.txt
  ↪  com/uofr/course/csc440/project/newspopularity/data/glove.twitter.27B.25d.txt
```

   Post that run the below mentioned command.

```
python3 -m com.uofr.course.csc440.project.newspopularity.runModels
 ↪  --train_test_split 0.2 --model nnet --log_interval 10 --batch_size
 ↪  100 --epochs 150 --lr 0.001 --seed 0 --dropout 0.35 --weight_decay
 ↪  0.001 --use_embedding
```

6. **Running Random Forest Classifier with optimized hyper parameters with top 36 attributes**

```
python3 -m com.uofr.course.csc440.project.newspopularity.runModels
 ↪  --train_test_split 0.2 --model forest --seed 0 --max_depth_forest
 ↪  26 --trees_forest 550 --k 36
```

7. **Extra Trees Classifier with optimized hyper parameters with top 36 attributes**

```
python3 -m com.uofr.course.csc440.project.newspopularity.runModels
 ↪  --train_test_split 0.2 --model etree --seed 0
 ↪  --max_depth_extra_trees 22 --trees_extra 500 --k 36
```