Introduction to Machine Learning (Spring 2019)

Homework #3 (50 Pts, May 22)

Student ID 2014310279

Name 김 승현

Instruction: We provide all codes and datasets in Python. Please write your code to complete the Evaluation metric. Compress 'Answer.py and submit with the filename 'HW3_STUDENT_ID.zip'.

(1) [20 pts] Implement four functions in 'utils/ Answer.py'. ('Accurcay,' 'Precision', 'Recall', 'F_measure',). You don't need to implement the part of model(Logistic Regression) and optimizer(SGD).

Answer: Fill your code here. You also have to submit your code to i-campus.

```
def Accuracy(label, pred):
    # ======================== EDIT HERE ========================
    Acc = None

    if len(pred.shape) == 1:
       pred = np.expand_dims(pred, 1)
    if len(label.shape) == 1:
        label = np.expand_dims(label, 1)

    TP = 0
    TN = 0

    for i in range(label.shape[0]):
        if(label[i][0] == 1 and pred[i][0] == 1):
            TP += 1
        elif(label[i][0] == 0 and pred[i][0] == 0):
            TN += 1

    Acc = (TP + TN)/ label.shape[0]
    # ===========================================================
```

```python
def Precision(label, pred):
    # ========================= EDIT HERE =========================
    precision = None
    if len(pred.shape) == 1:
       pred = np.expand_dims(pred, 1)
    if len(label.shape) == 1:
        label = np.expand_dims(label, 1)

    TP = 0
    FP = 0

    for i in range(label.shape[0]):
        if(label[i][0] == 1 and pred[i][0] == 1):
            TP += 1
        elif(label[i][0] == 0 and pred[i][0] == 1):
            FP += 1

    precision = TP / (TP + FP)
    # ============================================================
```

```python
def Recall(label, pred):
    # ========================= EDIT HERE =========================
    recall = None

    if len(pred.shape) == 1:
       pred = np.expand_dims(pred, 1)
    if len(label.shape) == 1:
        label = np.expand_dims(label, 1)

    TP = 0
    FN = 0

    for i in range(label.shape[0]):
        if(label[i][0] == 1 and pred[i][0] == 1):
            TP += 1
        elif(label[i][0] == 1 and pred[i][0] == 0):
            FN += 1
```

```
    recall = TP / (TP+FN)
    # ================================================================
```

```
def F_measure(label, pred):
    # ======================== EDIT HERE ========================
    F_score = None
    if len(pred.shape) == 1:
       pred = np.expand_dims(pred, 1)
    if len(label.shape) == 1:
        label = np.expand_dims(label, 1)

    TP = 0
    FP = 0
    FN = 0

    for i in range(label.shape[0]):
        if(label[i][0] == 1 and pred[i][0] == 1):
            TP += 1
        elif(label[i][0] == 0 and pred[i][0] == 1):
            FP += 1
        elif(label[i][0] == 1 and pred[i][0] == 0):
            FN += 1

    p = TP/(TP + FP)
    r = TP/(TP + FN)
    F_score = (2*p*r) / (p+r)
    # ================================================================
```

[NOTE]: You should write your codes in 'EDIT HERE' signs. It is not recommended to edit other parts. Once you complete your implementation, you can test the each evaluation metric (Accurcay, Precision, Recall, F-measure) for the given conditions with 'A_check.py'. You must get the same result with the examples.

[NOTE]: Dataset: These Datasets(Contracept, Heart and Yeast) are used for binary classifiacation. Each dataset's last column is the label(y), other part(x) are input_features.

(a) Show your codes for 'utils/Answers.py'.

(b) In 'A_main.py', you will deal with the 3 binary classification datasets, 'Contracept', 'Heart' and 'Yeast'. Label 1 is the positive and 0 is negative. With given hyperparameters, obtain 4 measures (accuracy, precision,

recall and F-measure) for 3 datasets and fill in the blank.

Answer: Fill the blank in the table.

Dataset	Contracept	Heart	Yeast
Accuracy	0.581633	0.766667	0.517857
Precision	0.640167	0.725000	0.333333
Recall	0.805263	0.906250	0.038462
F-measure	0.713287	0.805556	0.068966

Parameter Settings	
Batch size	32
Learning rate	0.01
Optimizer	SGD
# of epochs	100
Numpy Random_seed	503

<A_main.py parameter setting>

(2) [30 pts] Implement two functions in 'utils/Answer.py'. ('MAP' , 'nDCG'). You don't need to implement the part of model.

Answer: Fill your code here. You also have to submit your code to i-campus.

```
def MAP(label, hypo, at = 10):
    # ========================= EDIT HERE =========================
    Map = None
    argsorted_hypo = np.flip(np.argsort(hypo),1)
    sorted_label = np.zeros(label.shape)

    for i in range(label.shape[0]):
        for j in range(label.shape[1]):
            k = argsorted_hypo[i][j]
```

```
            sorted_label[i][j] = label[i][k]

    precision_np = np.zeros(label.shape)
    ap = []
    for i in range(label.shape[0]):
        rel_sum = 0
        for j in range(label.shape[1]):
            if sorted_label[i][j] == 1:
                rel_sum += 1
                precision_np[i][j] = rel_sum/(j+1)

        tmp_ap = 0
        for k in range(at):
            tmp_ap += precision_np[i][k]
        tmp_ap /= rel_sum
        ap.append(tmp_ap)

    Map = sum(ap) / label.shape[0]
    # ==============================================================
```

```
def nDCG(label, hypo, at = 10):
def DCG(label, hypo, at=10):
        # ======================== EDIT HERE ========================
        dcg = []
        argsorted_hypo = np.flip(np.argsort(hypo),1)
        sorted_label = np.zeros(label.shape)

        for i in range(label.shape[0]):
            for j in range(label.shape[1]):
                k = argsorted_hypo[i][j]
                sorted_label[i][j] = label[i][k]

        for i in range(label.shape[0]):
            tmp_dcg = 0
            for j in range(at):
                k = np.log2(j+2)
                if sorted_label[i][j] == 1:
                    tmp_dcg += 1/k
```

```
            dcg.append(tmp_dcg)


        # ============================================================
        return dcg

    def IDCG(label, hypo, at=10):
        # ========================= EDIT HERE =========================
        print(label.shape)
        idcg = []
        argsorted_hypo = np.flip(np.argsort(hypo),1)
        sorted_label = np.zeros(label.shape)
        for i in range(label.shape[0]):
            for j in range(label.shape[1]):
                k = argsorted_hypo[i][j]
                sorted_label[i][j] = label[i][k]

        idcg_label = -np.sort(-sorted_label)
        for i in range(label.shape[0]):
            tmp_idcg = 0
            for j in range(at):
                k = np.log2(j+2)
                if idcg_label[i][j] == 1:
                    tmp_idcg += 1/k
            idcg.append(tmp_idcg)

        # ============================================================
        return idcg
    # ========================= EDIT HERE =========================
    ndcg = 0
    dcg_list = DCG(label, hypo, at)
    idcg_list = IDCG(label, hypo, at)
    dcg_len = len(dcg_list)
    tmp_ndcg = 0
    for i in range(dcg_len):
        tmp_ndcg += (dcg_list[i])/ (idcg_list[i])
    ndcg = tmp_ndcg / dcg_len
    # ============================================================
    return ndcg
```

NOTE: You should write your codes in 'EDIT HERE' signs. It is not recommended to edit other parts. Once you complete your implementation, you can test the each evaluation metric (MAP, nDCG) for the given conditions with 'B_check.py'. You must get the same result with the examples.

(a) Show your codes for 'utils/Answers.py'. In 'B_check.py' you can test the each evaluation metric(MAP, nDCG) for the given conditions. You must get the same result for examples. (When you get the MAP, nDCG, You should get the mean value.)

(b) In 'B_main.py', you will deal with the 'Product.csv.', 'Movielens.csv' dataset. But you don't need to process this dataset. Prediction scores and correct labels are given. Test the datasets with MAP, nDCG at (25, 50, 75) and fill in the blanks.

Dataset	Product	Movielens
MAP @25	0.142387	0.337997
nDCG @25	0.495483	0.829796
MAP @50	0.269029	0.448933
nDCG @50	0.505574	0.805522
MAP @75	0.393560	0.510219
nDCG @75	0.660047	0.806849