15Z332 Ex6 - ANNMovieRating

October 17, 2018

1 Exercise 6

1.1 Artificial Neural Networks

Create a training model using ANN for the movie dataset. Class label is movie rating. Find the accuracy of your algorithm using 10-fold cross validation and leave-once cross validation

1.1.1 Step 1: Import Movie dataset

```
In [1]: import numpy as np
        import pandas as pd
        df = pd.read_csv("movie_metadata.csv")
        print(df.shape)
        df.head()
```

```
(5043, 28)
```

Out[1]:

	color	director_name	num_critic_for_reviews	duration	director_facebook_likes	actor_3_facebook_likes	actor_2_name	actor_1_facebook_likes	gross	genres
0	Color	James Cameron	723.0	178.0	0.0	855.0	Joel David Moore	1000.0	760505847.0	Action|Adventure|Fantasy|Sci-Fi
1	Color	Gore Verbinski	302.0	169.0	563.0	1000.0	Orlando Bloom	40000.0	309404152.0	Action|Adventure|Fantasy
2	Color	Sam Mendes	602.0	148.0	0.0	161.0	Rory Kinnear	11000.0	200074175.0	Action|Adventure|Thriller
3	Color	Christopher Nolan	813.0	164.0	22000.0	23000.0	Christian Bale	27000.0	448130642.0	Action|Thriller
4	NaN	Doug Walker	NaN	NaN	131.0	NaN	Rob Walker	131.0	NaN	Documentary

```
          ...          num_user_for_reviews language  country  content_rating  \
0         ...                        3054.0  English      USA           PG-13
1         ...                        1238.0  English      USA           PG-13
2         ...                         994.0  English       UK           PG-13
3         ...                        2701.0  English      USA           PG-13
4         ...                           NaN      NaN      NaN             NaN

        budget  title_year actor_2_facebook_likes imdb_score  aspect_ratio  \
0  237000000.0      2009.0                  936.0        7.9          1.78
1  300000000.0      2007.0                 5000.0        7.1          2.35
2  245000000.0      2015.0                  393.0        6.8          2.35
3  250000000.0      2012.0                23000.0        8.5          2.35
4          NaN         NaN                   12.0        7.1           NaN

  movie_facebook_likes
0                33000
1                    0
2                85000
3               164000
4                    0

[5 rows x 28 columns]
```

1.1.2 Step 2: Feature selection

For this neural network, the attributes selected for training the model are: * 'num_critic_for_reviews' * 'director_facebook_likes' * 'actor_1_facebook_likes' * 'actor_2_facebook_likes' * 'actor_3_facebook_likes' * 'movie_facebook_likes'

The target attribute is imdb_score.

```
In [2]: df = df[['num_critic_for_reviews','director_facebook_likes','actor_1_facebook_likes','a
        print(df.shape)
        df.head()
```

```
(5043, 7)
```

```
Out[2]:    num_critic_for_reviews  director_facebook_likes  actor_1_facebook_likes  \
0                   723.0                      0.0                  1000.0
1                   302.0                    563.0                 40000.0
2                   602.0                      0.0                 11000.0
3                   813.0                  22000.0                 27000.0
4                     NaN                    131.0                   131.0

   actor_2_facebook_likes  actor_3_facebook_likes  movie_facebook_likes  \
0                   936.0                   855.0                 33000
1                  5000.0                  1000.0                     0
2                   393.0                   161.0                 85000
```

```
3                 23000.0                 23000.0                 164000
4                    12.0                     NaN                      0

   imdb_score
0         7.9
1         7.1
2         6.8
3         8.5
4         7.1
```

1.1.3 Step 3: Handling missing data

First we will check the number of missing values in each attribute and the type of attribute they are (numerical or categorical).

```
In [3]: df.isnull().sum()

Out[3]: num_critic_for_reviews      50
        director_facebook_likes    104
        actor_1_facebook_likes       7
        actor_2_facebook_likes      13
        actor_3_facebook_likes      23
        movie_facebook_likes         0
        imdb_score                   0
        dtype: int64

In [4]: df.dtypes

Out[4]: num_critic_for_reviews     float64
        director_facebook_likes    float64
        actor_1_facebook_likes     float64
        actor_2_facebook_likes     float64
        actor_3_facebook_likes     float64
        movie_facebook_likes         int64
        imdb_score                 float64
        dtype: object
```

Since all the attributes are numerical data type, we fill the missing values with mean or median. Here, they are filled with mean values

```
In [5]: for i in df.columns:
            if df[i].isnull().sum()>0:
                df[i] = df[i].fillna(df[i].mean())
        df.isnull().sum()

Out[5]: num_critic_for_reviews     0
        director_facebook_likes    0
        actor_1_facebook_likes     0
        actor_2_facebook_likes     0
```

```
actor_3_facebook_likes     0
movie_facebook_likes       0
imdb_score                 0
dtype: int64
```

1.1.4 Step 4: Model ANN and predict

Create an object for MLPClassifier from sklearn package and split the entire dataset into Y (only target attribute) and X (all attributes except target).

```
In [6]: from sklearn.neural_network import MLPClassifier

        Y = df['imdb_score'].round().values
        X = df.drop(['imdb_score'],axis=1).values

        clf = MLPClassifier(hidden_layer_sizes=(5), solver='sgd', activation="logistic")
```

1) Using 10-fold cross validation, create the ANN model and compute the accuracy for prediction

```
In [7]: from sklearn.model_selection import KFold
        kf = KFold(n_splits=10)
        for train_indices, test_indices in kf.split(X):
            clf.fit(X[train_indices], Y[train_indices])
            print(clf.score(X[test_indices], Y[test_indices]))
```

```
0.3702970297029703
0.401980198019802
0.29306930693069305
0.3551587301587302
0.32341269841269843
0.34325396825396826
0.375
0.2996031746031746
0.2757936507936508
0.2896825396825397
```

2) Using leave-once validation, create the ANN model and compute the accuracy for prediction

```
In [36]: from sklearn.model_selection import LeaveOneOut
         loo = LeaveOneOut()

         Y = df['imdb_score'].round()[0:100].values
         X = df.drop(['imdb_score'],axis=1)[0:100].values
         avg=0
         for train_indices, test_indices in loo.split(X):
             clf.fit(X[train_indices], Y[train_indices])
             avg = avg + clf.score(X[test_indices], Y[test_indices])
         print('Average accuracy score: ',avg/100)
```

```
/usr/local/lib/python3.7/site-packages/sklearn/neural_network/multilayer_perceptron.py:564: Co
  % self.max_iter, ConvergenceWarning)


Average accuracy score:  0.31
```