Big Data Analytics - Individual Project

By Muhammad Haris Choudhary

Movie Ratings and Recommendations

Pre-processing

Requirements

- Name -> length
- Separate Genre
- Remove Unknowns from episodes
- Changing Length to int
- Moving each item back to place
- Lastly, scaling

Procedure

1. Used Pandas, numpy
2. A function called preProcess()
3. Returns 2 ndarrays
4. X and Y values depending on the file name.

Task 1

Predict Ratings for each new telefilm

Procedure

Libraries

Numpy

Tensorflow

Keras

Plans

- Linear Regression
- Polynomial Regression
- Neural Network
- Tensorflow

```
ral(X, Y, X_pred):

el = keras.Sequential([
 keras.layers.Dense(7, activation=tf.nn.relu),
 keras.layers.Dense(64, activation=tf.nn.relu),
 keras.layers.Dense(64, activation=tf.nn.relu),
 keras.layers.Dense(1)


el.compile(optimizer='adam',
           loss='mean_squared_error',
           metrics=tf.keras.metrics.RootMeanSquaredError())

el.fit(X.astype('float32'), Y.astype('float32'), epochs=1000, verbose=0)

d_y = model.predict(X_pred.astype('float32'))

urn pred_y
```

```
class NeuralNetwork(object):
    def __init__(self, x, y):
        #parameters
        np.random.seed(0)
        self.input  = x
        self.size   = x.shape[1]
        self.w1     = 0.1 * np.random.rand(int(x.shape[1]), 89)
        self.w2     = 0.1 * np.random.rand(89, 1)
        self.y      = y
        self.output = np.zeros(y.shape)

    def feedForward(self):
        #forward propogation through the network
        self.l1 = np.dot(self.input, self.w1)
        self.layer1 = sigmoid(self.l1) #Layer 1
        self.output = sigmoid(np.dot(self.layer1, self.w2)) #Output


    def backprop(self):
        self.output_error = self.y - self.output # error in output
        self.output_delta = self.output_error * sigmoid(self.output, deriv=True)

        self.z2_error = self.output_delta.dot(self.w2.T) #z2 error: how much our hidden layer w
                                                                         ng derivative o

                                                                         -> hidden) weig
                                                                         hidden -> outpu
```

```
def Poly():
    X, y = preprocess.preProcess()
    y = np.reshape(y, (X.shape[0],1))

    alpha = 0
    deg = X.shape[1]

    model = Ridge(alpha=alpha, solver='auto', random_state = 42)
    model = Pipeline([
            ("poly_features", PolynomialFeatures(degree=deg, include_bias=True)),
            ("std_scaler", StandardScaler()),
            ("regul_reg", model),
        ])
    model.fit(X, y)
    #y_pred = model.predict(x_test)
```

Results

Task 2

Recommender System

Procedure

Libraries

pyspark

Plans

- Content based

  Not as effective

- Collaborative-Filtering

  Sufficient data

```
#Reading from the file

ratings_df = spark.read.csv("Rating.csv", header=True, inferSchema=True)

ratings_df = ratings_df.where("rating != -1")

#initialzing recommender
recommender = ALS(maxIter=25, regParam=0.1, userCol="user_id",
                  itemCol = "teleplay_id", ratingCol = "rating",
                  coldStartStrategy = "drop")

train, test = ratings_df.randomSplit([0.9, 0.1])

trained = recommender.fit(train)

prediction = trained.transform(test)

prediction.show()

# Create a user for recommending
user_input = ratings_df.select("user_id").where(col("user_id") == 53698)

user = trained.recommendForUserSubset(user_input, 100000)

user = user.withColumn("rec_exp", explode("recommendations")).select('user_id',
                        col("rec_exp.teleplay_id"),
                        col("rec_exp.rating"))

user.show(10, False)
```

user_id	teleplay_id	rating	prediction
5214	148	6	6.377176
51013	148	6	6.993211
73135	148	3	5.393895
59186	148	6	7.3444896
33175	148	8	6.961884
66105	148	5	6.31864
66563	148	7	6.2818904
59311	148	7	7.57663
47582	148	9	7.2505164
58646	148	8	6.721721
50558	148	5	6.1753287
15629	148	4	6.0460157
50766	148	6	6.850387
53640	148	9	7.2064137
10744	148	8	6.0731225
30587	148	6	7.1608324
58570	148	4	6.843752
59172	148	8	7.872093
47935	463	10	8.503885
58944	463	8	7.867219

Results