

READY

61130700310 อภิชา สীগเมน  
61130700318 ดนุศิษฐ์ อริยกุลเจษฎางค์  
61130700332 นรเศรษฐี พจนศึกษาภล

FINISHED

```
--2020-04-26 12:28:29-- https://github.com/0sirisz/gamesaledataset/raw/master/game_sale_dataset.csv
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/0sirisz/gamesaledataset/master/game_sale_dataset.csv [following]
--2020-04-26 12:28:29-- https://raw.githubusercontent.com/0sirisz/gamesaledataset/master/game_sale_dataset.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.0.133, 151.101.64.133, 151.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.0.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1006329 (983K) [text/plain]
Saving to: 'game sale dataset.csv.1'
```

0K	.....	5%	2.22M	0s
50K	.....	10%	5.55M	0s
100K	.....	15%	14.1M	0s

Took 6 sec. Last updated by anonymous at April 26 2020, 7:28:31 PM.

READY

โค้ดสำหรับดาวน์โหลด dataset โดยจะทำการตรวจสอบก่อน ไฟล์ข้อมูลที่ต้องการนั้นมีอยู่ใน hdfs แล้วหรือยัง หากยังไม่มี จะทำการโ

 SPARK JOBS FINISHED

```
root
|-- name: string (nullable = true)
|-- platform: integer (nullable = true)
|-- year: integer (nullable = true)
|-- genre: integer (nullable = true)
|-- publisher: integer (nullable = true)
|-- na: double (nullable = true)
|-- eu: double (nullable = true)
|-- jp: double (nullable = true)
|-- other: double (nullable = true)
|-- global_sales: double (nullable = true)
|-- label: integer (nullable = true)
```

name	platform	year	genre	publisher	na	eu	jp	other	global_sales	label
Wii Sports	26	2006	11	368	41.49	29.02	3.77	8.46	82.74	1
Super Mario Bros.	12	1985	5	268	10.08	2.50	6.81	0.77	19.16	1

## gameSale

Took 16 sec. Last updated by anonymous at May 03 2020, 5:44:18 PM.

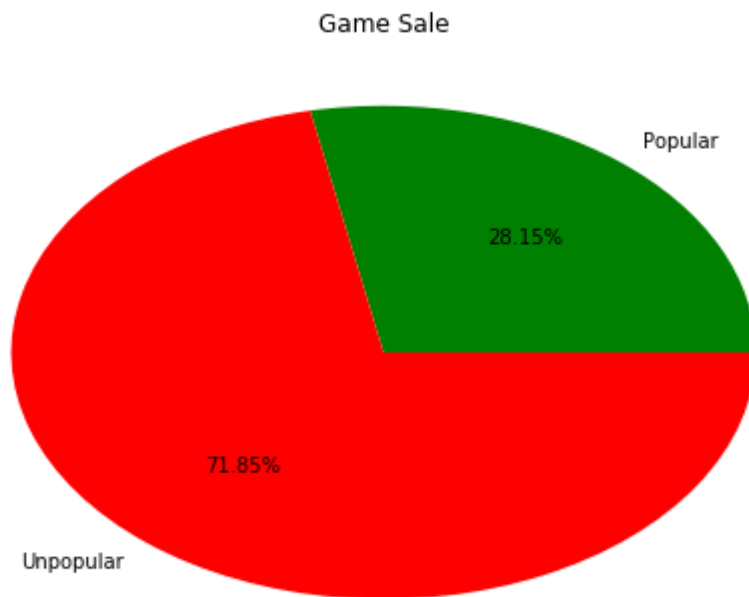
### Description

READY

โค้ดสำหรับใช้แสดงข้อมูลจากไฟล์ csv ที่ดาวน์โหลดมาแล้วข้างต้น และแสดงข้อมูลออกมาเป็นตาราง โดยเก็บข้อมูลเอาไว้ในตัวแปรที่ชื่อ

### Visualization

FINISHED



Took 1 sec. Last updated by anonymous at April 26 2020, 7:28:33 PM. (outdated)

### Description

READY

โค้ดสำหรับฟิวเดอร์ข้อมูล โดยกรองข้อมูลเฉพาะแอททริบิวต์ที่ชื่อว่า label โดยจะมีข้อมูลอยู่สองค่า คือ Popular ซึ่งแสดงข้อมูลด้วย 1 จากนั้นทำการ count ข้อมูล 1 และ 0 ว่ามีจำนวนทั้งหมดเท่าไร และพล็อตออกมาเป็น pie chart

### Create hive table with hive(Option 1 must run in hive)

READY

### Description

READY

(ตัวเลือกที่ 1)  
โค้ดสำหรับสร้าง hive แล้วเก็บข้อมูลลงใน path ที่กำหนด เพื่อเป็น input path สำหรับดาวน์โหลดข้อมูลต่อไป

## Create hive table with spark(Option 2)

# gameSale

DataFrame[]

FINISHED

Took 1 sec. Last updated by anonymous at April 26 2020, 7:28:34 PM.

### Description

READY

(ตัวเลือกที่ 2)

โค้ด pyspark สำหรับสร้าง Temp Table แล้วทำการนำข้อมูลจากตัวแปรที่ชื่อ data แล้วเก็บลงไว้ใน Temp Table จากนั้นจึงทำการส

### Get data from hive table

Spark Job ID (http://data-2020-master-class-m.us-central1-a.c.mythic-hulling-244908.internal:4040/jobs/job?id=3)

FINISHED

	etris	6 1989	6	368	23.2	2.26	4.22	0.58	30.26	1
	New Super Mario B...	5 2006	5	368	11.38	9.23	6.5	2.9	30.01	1
	Wii Play	26 2006	4	368	14.03	9.2	2.93	2.85	29.02	1
	New Super Mario B...	26 2009	5	368	14.59	7.06	4.7	2.26	28.62	1
	Duck Hunt	12 1984	9	368	26.93	0.63	0.28	0.47	28.31	1
	Nintendogs	5 2005	10	368	9.07	11.0	1.93	2.75	24.76	1
	Mario Kart DS	5 2005	7	368	9.81	7.57	4.13	1.92	23.42	1
	Pokemon Gold/Poke...	6 1999	8	368	9.0	6.18	7.2	0.71	23.1	1
	Wii Fit	26 2007	11	368	8.94	8.03	3.6	2.15	22.72	1
	Wii Fit Plus	26 2009	11	368	9.09	8.59	2.53	1.79	22.0	1
	Kinect Adventures!	29 2010	4	328	14.97	4.94	0.24	1.67	21.82	1
	Grand Theft Auto V	18 2013	1	498	7.01	9.27	0.97	4.14	21.4	1
	Grand Theft Auto:...	17 2004	1	498	9.43	0.4	0.41	10.57	20.81	1
	Super Mario World	24 1990	5	368	12.78	3.75	3.54	0.55	20.61	1
	Brain Age: Train ...	5 2005	4	368	4.75	9.26	4.16	2.05	20.22	1

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

only showing top 20 rows

Took 2 sec. Last updated by anonymous at May 03 2020, 5:44:39 PM.

### Description

READY

โค้ดสำหรับดึงข้อมูลจาก hive table ขึ้นมาใช้

### Detail of data

FINISHED

[Row(mean=0.5374406555006629, std=1.555027935569911)]

Took 0 sec. Last updated by anonymous at April 26 2020, 7:28:34 PM. (outdated)

### Description

ERROR

โค้ดสำหรับใช้หาค่า mean และ standard deviation (std)

Took 0 sec. Last updated by anonymous at May 03 2020, 5:53:02 PM. (outdated)

## Split train-test set gameSale

SPARK JOBS FINISHED



settings ▾

name ▾	platform ▾	year ▾	genre ▾	≡p
007: Tomorrow Never Dies	16	1999	9	1
2 Games in 1: Disney Princess & The Lion King	7	2004	4	5
2 Games in 1: Sonic Advance & ChuChu Rocket!	7	2005	4	5
50 Cent: Bulletproof	17	2005	1	5
A Run's Life	16	1998	5	4

Output is truncated to 1000 rows. Learn more about `zeppelin.spark.maxResult`



Took 2 sec. Last updated by anonymous at May 03 2020, 5:45:39 PM.

### Description

READY

โค้ดสำหรับการแบ่งข้อมูลออกเป็นสองส่วน เพื่อใช้ในการ train และใช้ในการ test โดยอัตราส่วนของข้อมูลระหว่าง train ต่อ test

### Transformers and Parameters with Neural network

FINISHED

Took 0 sec. Last updated by anonymous at May 03 2020, 6:07:38 PM.

### Description

READY

โค้ดสำหรับปรับแต่งโมเดล โดยทำการปรับแต่งค่าพารามิเตอร์ต่างๆ ดังต่อไปนี้

- combineFeature โดยการคัดเลือกเอาหัวแตรที่นำมาใช้ในการทำนาย
- nn โดยใช้อัลกอริทึม MultilayerPerceptron ในการสร้างโมเดล ทำทั้งหมด 10 รอบ
- paramGrid โดยโมเดลแรก มีเพียง input 5 nodes และ output 2 nodes โดยที่ไม่มี hidden layers

จากนั้น นำข้อมูลทั้งหมดไปเก็บไว้ในตัวแปรที่ชื่อว่า nn\_pipeline

### Model Training

SPARK JOBS FINISHED

Took 17 sec. Last updated by anonymous at May 03 2020, 6:05:21 PM.

Description

คำสั้งสำหรับการ train โมเดลที่หนึ่ง ด้วย training set

gameSale

READY

```
%pyspark
## auto hyperparameter tuning
print("Num of weights = %d"%len(nn_model.stages[2].bestModel.weights))
nn_model.stages[2].bestModel.weights
```

Num of weights = 12  
DenseVector([-0.6994, -0.597, -0.4993, -0.543, -0.8095, -0.7629, -0.4966, -0.4965, -7.871, 7.5253, 0.165, 0.7614])

Took 0 sec. Last updated by anonymous at May 03 2020, 6:05:22 PM.

FINISHED

Description

โมเดลที่หนึ่ง จำนวนโหนด มีเท่ากับ 12 โหนด

READY

Test

Took 0 sec. Last updated by anonymous at May 03 2020, 6:05:26 PM.

FINISHED

Description

คำสั้งสำหรับการ test โมเดลที่หนึ่ง

READY

download

▼

settings ▼

SPARK JOBS FINISHED

name ▼	platform ▼	year ▼	genre ▼	published ▼
2 Games in 1 Double Pack: The Incredibles / Finding Nemo: The Continuing Adventures	7	2007	1	514
ATV Offroad Fury 2	17	2002	7	514

Output is truncated to 102400 bytes. Learn more about ZEPPELIN\_INTERPRETER\_OUTPUT\_LIMIT



# gameSale

Took 1 sec. Last updated by anonymous at May 03 2020, 6:05:30 PM.

## Description

READY

คำสั่งแสดงผลลัพธ์ที่ได้จากโมเดลที่หนึ่ง

## Result

SPARK JOBS FINISHED

```
('Accuracy', 0.9662650602409638)
('F1', 0.9655542625186935)
('Root Mean Squared Error (RMSE)', 0.18367073735093498)
```

Took 2 sec. Last updated by anonymous at May 03 2020, 6:08:35 PM. (outdated)

## Description

READY

โค้ดสำหรับวัดประสิทธิภาพของผลการทำนายที่ได้จากโมเดลที่หนึ่ง  
โดยมีค่า Accuracy เท่ากับ 96.63%  
มีค่า F1 เท่ากับ 96.55%  
และมีค่า RMSE เท่ากับ 18.37%

SPARK JOBS FINISHED

```
Confusion matrix:
DenseMatrix([[4803.,    0.],
              [ 224., 1613.]])
```

Took 4 sec. Last updated by anonymous at May 03 2020, 6:08:51 PM.

## Description

READY

โค้ดสำหรับแสดงผล confusion matrix สำหรับโมเดลที่หนึ่ง

## Transformers and Parameters with Neural network2

FINISHED

Took 0 sec. Last updated by anonymous at May 03 2020, 6:09:07 PM. (outdated)

## Description

READY

โค้ดสำหรับปรับแต่งโมเดล โดยทำการปรับแต่งค่าพารามิเตอร์ต่างๆ ดังต่อไปนี้

- combineFeature โดยการคัดเลือกเอาหัวแตรที่มีตัวมาใช้ในการทำนาย
- nn โดยใช้อัลกอริทึม MultilayerPerceptron ในการสร้างโมเดล ทำทั้งหมด 10 รอบ
- paramGrid โดยโมเดลที่สอง มี input 5 nodes, output 2 nodes และมี hidden layers ทั้งหมด 3 layers [10,10,3]

จากนั้น นำข้อมูลที่ทำมาได้ทั้งหมดไปเก็บไว้ในตัวแปรที่ชื่อว่า nn\_pipeline

```
%pyspark
# Estimator
nn_model2 = nn_pipeline2.fit(trainingSet)
```

Took 14 sec. Last updated by anonymous at May 03 2020, 6:09:26 PM. (outdated)

## Description

READY

คำสั่งสำหรับการ train โมเดลที่สอง ด้วย training set

```
%pyspark
## auto hyperparameter tuning
print("Num of weights = %d"%len(nn_model2.stages[2].bestModel.weights))
nn_model2.stages[2].bestModel.weights
```

FINISHED

Num of weights = 211

DenseVector([-0.4783, -0.8183, -0.2368, -0.8075, -0.8742, -0.7126, -0.2309, -0.7741, 0.29, -0.64, -0.0983, 1.0209, 1.0178, -0.194, -1.073, 0.5703, -0.3067, 0.4594, 0.5292, -0.7449, -0.4391, 0.0488, -0.4438, -0.7689, -0.2055, -0.15, 0.729, 0.0167, -0.1066, -0.2618, -0.9678, -0.8347, -1.0031, -0.6169, -0.4926, 0.4611, 0.6461, -0.6491, 0.4688, -0.346, -1.0209, -0.7873, 0.4844, 0.4544, -0.1137, 0.588, 0.6474, 0.0111, 0.5098, 0.9569, -0.8823, -0.6937, 0.8694, 0.9883, 0.4676, 0.3672, -0.9459, 0.1971, 0.4817, 0.8857, 0.3305, -0.6163, 0.4179, -0.615, 0.5403, 0.7864, 0.6466, -0.5548, 0.0216, 0.2839, 0.5176, 0.1185, -0.1538, 0.0085, -0.3363, 0.3573, -0.1587, -0.5905, 0.301, -0.2461, -0.0496, -0.7395, 0.3313, 0.3411, 0.3119, 0.3953, -0.6041, -0.3307, -0.2675, -0.307, -0.5536, 0.4242, -0.4481, 0.7296, -0.2709, 0.4438, 0.2571, 0.3696, -0.0463, 0.6149, 0.1972, -0.4365, -0.5996, -0.6473, -0.1569, -0.072, -0.6882, 0.1483, 0.0076, -0.4642, -0.3365, 0.4289, 0.5794, -0.0932, -0.4635, -0.1027, 0.3124, 0.1021, 0.0074, -0.6383, 0.6869, -0.184, -0.4515, 0.7504, -0.656, 0.2135, 0.4988, -0.2318, -0.069, 0.1095, 0.5038, 0.3675, -0.1483, -0.622, 0.561, 0.5013, 0.1244, 0.6486, 0.2211, -0.2703, 0.6341, 0.3963, 0.3929, 0.3449, 0.5843, 0.3531, 0.324, 0.0678, -0.3828, -0.358, -0.4357, -0.5116, -0.4333, -0.6275, -0.1201, -0.553, -0.5558, 0.2281, 0.4304, 0.5033, -0.6635, 0.234, -0.0992, 0.5549, 0.2667, 0.2065, 0.1109, -0.2022, -0.1644, -0.3254, -0.1757, 0.3197, -0.6918, 0.3808, 0.7546, 0.0376, -0.5065, 0.0929, 0.4102, 0.8483, 0.544, 0.185, -0.2026, -0.2986, -0.3161, -0.6528, 0.1803, -0.6058, -0.6094, -0.591, -0.1088, 0.1583, 0.5978, -0.3774, -0.7607, -0.5786, -0.7837, -0.1365, -0.5588, -0.7717, 0.0472])

Took 0 sec. Last updated by anonymous at May 03 2020, 6:09:46 PM.

## Description

READY

โมเดลที่สอง จำนวนโหนด มีเท่ากับ 211 โหนด  
หากมีการเพิ่มจำนวน hidden layers มากขึ้น จำนวนโหนดก็ย่อมเพิ่มมากขึ้นด้วย

## Test

FINISHED

```
%pyspark
# Transformers
nn_prediction2 = nn_model2.transform(testSet)
```

Took 0 sec. Last updated by anonymous at May 03 2020, 6:09:57 PM.

## Description

READY

คำสั่งสำหรับการ test โมเดลที่สอง

# gameSale

```
%pyspark
z.show(nn_prediction2)
```

SPARK JOBS FINISHED



name	platform	year	genre	publisher
1080°: TenEighty Snowboarding	11	1998	11	368
2 Games in 1 Double Pack: The Incredibles / Finding Nemo: The Continuing Adventures	7	2007	1	514

Output is truncated to 102400 bytes. Learn more about ZEPPELIN\_INTERPRETER\_OUTPUT\_LIMIT



Took 1 sec. Last updated by anonymous at May 03 2020, 6:10:02 PM. (outdated)

## Description

READY

คำสั่งแสดงผลลัพธ์ที่ได้จากโมเดลที่สอง

## Result

SPARK JOBS FINISHED

```
%pyspark
nn_evaluator2 = MulticlassClassificationEvaluator(predictionCol="prediction", labelCol="label")
nn_rmse_evaluator2 = RegressionEvaluator(labelCol="label", predictionCol="prediction", metricName="rms")
print("Accuracy",nn_evaluator2.evaluate(nn_prediction2, {nn_evaluator2.metricName: "accuracy"}))
print("F1",nn_evaluator2.evaluate(nn_prediction2, {nn_evaluator2.metricName: "f1"}))
print("Root Mean Squared Error (RMSE)",nn_rmse_evaluator2.evaluate(nn_prediction2, {nn_rmse_evaluator2.metricName: "rmse"}))

('Accuracy', 0.7233433734939759)
('F1', 0.6072215717716624)
('Root Mean Squared Error (RMSE)', 0.5259815838088099)
```

Took 2 sec. Last updated by anonymous at May 03 2020, 6:10:08 PM. (outdated)



## Description

READY

gameSale

โค้ดสำหรับวัดประสิทธิภาพของผลการทำนายที่ได้จากโมเดลที่สอง  
โดยมีค่า Accuracy เท่ากับ 72.33%  
และค่า RMSE เท่ากับ 52.59%

```
%pyspark
from pyspark.mllib.evaluation import MulticlassMetrics
nn_metrics2 = MulticlassMetrics(nn_prediction2[["prediction","label"]].rdd.map(lambda x:[float(x[0]),f

# Confusion matrix
print("Confusion matrix:")
print(nn_metrics2.confusionMatrix())

Confusion matrix:
DenseMatrix([[4803.,    0.],
              [1837.,    0.]])
```

Took 2 sec. Last updated by anonymous at May 03 2020, 6:10:13 PM.

SPARK JOBS FINISHED

## Description

READY

โค้ดสำหรับแสดงผล confusion matrix สำหรับโมเดลที่สอง

## Transformers and Parameters with Neural network3

FINISHED

```
%pyspark
from pyspark.ml import Pipeline
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.ml.tuning import ParamGridBuilder, TrainValidationSplit
from pyspark.ml.classification import MultilayerPerceptronClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Transformers
combineFeature = VectorAssembler(
    inputCols=["platform", "year", "genre", "publisher", "global_sales"],
    outputCol="combinedFeatures")

# Transformers
scaler = StandardScaler(inputCol="combinedFeatures", outputCol="features",
                        withStd=True, withMean=True)

# Transformers
nn = MultilayerPerceptronClassifier(maxIter=20)

## 1 manual hyperparameter tuning
#nn.layers=[5,8,4,2]
# # Pipeline
# pipeline = Pipeline(stages=[combineFeature,scaler, nn])

##2 auto model selection and hyperparameter tuning
paramGrid = ParamGridBuilder()\
    .addGrid(nn.layers, [[5,10,10,10,10,3]])\
    .build()

tvs = TrainValidationSplit(estimator=nn,
                          estimatorParamMaps=paramGrid,
                          evaluator=MulticlassClassificationEvaluator(metricName="f1"),
                          # 80% of the data will be used for training, 20% for validation.
                          trainRatio=0.8)
```

```
Pipeline
nn_pipeline3 = Pipeline(stages=[combineFeature, scaler, tvs])
```

Took 0 sec. Last updated by anonymous at May 03 2020, 6:13:50 PM. (outdated)

## gameSale

### Description

READY

โค้ดสำหรับปรับแต่งโมเดล โดยทำการปรับแต่งค่าพารามิเตอร์ต่างๆ ดังต่อไปนี้

- combineFeature โดยการคัดเลือกเอาหัวแอมป์มาใช้ในการทำนาย
- nn โดยใช้อัลกอริทึม MultilayerPerceptron ในการสร้างโมเดล ทำทั้งหมด 20 รอบ
- paramGrid โดยโมเดลที่สาม มี input 5 nodes, output 2 nodes และมี hidden layers ทั้งหมด 4 layers [10,10,10,

จากนั้น นำข้อมูลทั้งหมดที่ได้ทั้งหมดไปเก็บไว้ในตัวแปรที่ชื่อว่า nn\_pipeline

### Model Training

SPARK JOBS FINISHED

```
%pyspark
# Estimator
nn_model3 = nn_pipeline3.fit(trainingSet)
```

Took 11 sec. Last updated by anonymous at May 03 2020, 6:14:05 PM.

### Description

READY

คำสั่งสำหรับการ train โมเดลที่สาม ด้วย training set

```
%pyspark
## auto hyperparameter tuning
print("Num of weights = %d"%len(nn_model3.stages[2].bestModel.weights))
nn_model3.stages[2].bestModel.weights
```

FINISHED

```
Num of weights = 412
DenseVector([-0.4575, -0.8384, -0.2481, -0.7949, -0.8964, -0.676, -0.2075, -0.7846, 0.2965, -0.6418,
-0.1611, 1.0384, 1.0383, -0.1977, -1.0362, 0.5203, -0.3328, 0.4736, 0.5129, -0.7452, -0.4218, 0.026
6, -0.4488, -0.7626, -0.2242, -0.1298, 0.7266, 0.0076, -0.1007, -0.2623, -0.9822, -0.8218, -1.0016,
-0.618, -0.4918, 0.4563, 0.6319, -0.6466, 0.4675, -0.3444, -0.9079, -0.8796, 0.4333, 0.4953, -0.277
5, 0.7022, 0.7402, -0.0602, 0.5479, 0.9505, -0.9312, -0.6578, 0.8726, 0.9941, 0.4771, 0.3496, -0.92
9, 0.1929, 0.4735, 0.8838, 0.3335, -0.5955, 0.2829, -0.52, 0.6319, 0.7415, 0.5265, -0.5662, 0.0901,
0.1926, 0.5237, 0.1461, -0.3425, 0.1408, -0.2022, 0.2887, -0.3285, -0.6078, 0.4008, -0.3737, -0.056
8, -0.7397, 0.3195, 0.3724, 0.3137, 0.4263, -0.6141, -0.335, -0.2537, -0.2973, -0.5632, 0.4201, -0.4
305, 0.7366, -0.2943, 0.495, 0.28, 0.3693, -0.0533, 0.6516, 0.1946, -0.4302, -0.6612, -0.602, -0.119
2, -0.0665, -0.7312, 0.1416, 0.0396, -0.4872, -0.3469, 0.4242, 0.6188, -0.0953, -0.4976, -0.0552, 0.
347, 0.1046, -0.0104, -0.5926, 0.6753, -0.2022, -0.3296, 0.6832, -0.7498, 0.2895, 0.6128, -0.2214, -
0.1323, 0.2126, 0.5024, 0.377, -0.2267, -0.5588, 0.6127, 0.4937, 0.0587, 0.6398, 0.2655, -0.31, 0.62
33, 0.3887, 0.445, 0.3335, 0.5401, 0.4042, 0.3712, 0.0712, -0.4056, -0.303, -0.4488, -0.5278, -0.340
9, -0.6685, -0.1941, -0.4692, -0.465, 0.2334, 0.3869, 0.5931, -0.6745, 0.2378, -0.1272, 0.5993, 0.27
39, 0.2411, 0.0931, -0.2055, -0.1477, -0.3156, 0.1506, 0.1643, -0.6959, 0.4294, 0.7183, 0.0369, -0.4
633, 0.1097, 0.4261, 0.6751, 0.6087, 0.1793, 0.266, -0.5073, -0.3518, -0.3463, 0.0738, -0.5939, -0.6
968 -0.5177 -0.2092 0.2576 0.6875 -0.2716 0.0107 -0.7071 -0.2801 -0.6110 -0.4681 -0.2617
```

Took 0 sec. Last updated by anonymous at May 03 2020, 6:13:24 PM.

### Description

READY

โมเดลที่สาม จำนวนโหนด มีเท่ากับ 412 โหนด

หากมีการเพิ่มจำนวน hidden layers มากขึ้น จำนวนโหนดก็ย่อมเพิ่มมากขึ้นด้วย

## Test gameSale

FINISHED

```
%pyspark
# Transformers
nn_prediction3 = nn_model3.transform(testSet)
```

Took 0 sec. Last updated by anonymous at May 03 2020, 6:14:07 PM.

## Description

READY

คำสั่งสำหรับการ train โมเดลที่สาม ด้วย training set

```
%pyspark
z.show(nn_prediction3)
```

SPARK JOBS FINISHED



settings ▼

name ▼	platform ▼	year ▼	genre ▼	published ▼
Call of Duty: Black Ops	18	2010	9	17
Call of Duty: Black Ops	29	2010	9	17

Output is truncated to 102400 bytes. Learn more about ZEPPELIN\_INTERPRETER\_OUTPUT\_LIMIT



Took 0 sec. Last updated by anonymous at May 03 2020, 6:13:30 PM.

## Description

READY

คำสั่งแสดงผลลัพธ์ที่ได้จากโมเดลที่สาม

## Result

SPARK JOBS FINISHED

```
%pyspark
nn_evaluator3 = MulticlassClassificationEvaluator(predictionCol="prediction", labelCol="label")
```

# gameSale

```
nn_rmse_evaluator3 = RegressionEvaluator(labelCol="label", predictionCol="prediction", metricName="rmse")
print("Accuracy",nn_evaluator3.evaluate(nn_prediction3, {nn_evaluator3.metricName: "accuracy"}))
print("F1",nn_evaluator3.evaluate(nn_prediction3, {nn_evaluator3.metricName: "f1"}))

('Accuracy', 0.869277108433735)
('F1', 0.8639295347)
('Root Mean Squared Error (RMSE)', 0.3615562080317043)
```

Took 1 sec. Last updated by anonymous at May 03 2020, 6:14:13 PM.

## Description

READY

โค้ดสำหรับวัดประสิทธิภาพของผลการทำนายที่ได้จากโมเดลที่สาม  
โดยมีค่า Accuracy เท่ากับ 86.93%  
มีค่า F1 เท่ากับ 86.39%  
และมีค่า RMSE เท่ากับ 36.16%

```
%pyspark
from pyspark.mllib.evaluation import MulticlassMetrics
nn_metrics3 = MulticlassMetrics(nn_prediction3[["prediction","label"]].rdd.map(lambda x:[float(x[0]),float(x[1])]))

# Confusion matrix
print("Confusion matrix:")
print(nn_metrics3.confusionMatrix())
```

SPARK JOBS FINISHED

Confusion matrix:  
DenseMatrix([[4568., 235.],  
[ 633., 1204.]])

Took 1 sec. Last updated by anonymous at May 03 2020, 6:14:20 PM.

## Description

READY

โค้ดสำหรับแสดงผล confusion matrix สำหรับโมเดลที่สาม

## Insert prediction result to hive table

SPARK JOB (http://big-data-2020-master-class-m.us-central1-a.c.mythic-hulling-244908.internal:4040/jobs/job?id=570)

FINISHED

DataFrame[]

Took 4 sec. Last updated by anonymous at May 03 2020, 6:17:05 PM.

## Description

READY

โค้ดสำหรับนำผลลัพธ์จากโมเดลที่ดีที่สุด ซึ่งก็คือโมเดลหนึ่งที่มีค่า Accuracy มากถึง 96.63%  
ไปใส่ไว้ใน hive table