UJIAN AKHIR SEMESTER

LAPORAN BIG DATA

(CLASSIFICATION REGRESSION)

Disusun oleh:

Ravino Rahman	2018420009
Muhammad Hafizh Azzasafah	2018420017
Febrian Dimas Winaputra	2018420064

UNIVERSITAS DR. SOETOMO SURABAYA

TAHUN AJARAN 2021-2022

Jenis Soal: Classification

Dataset: Starbucks Customer Survey

Link dataset: [link dataset starbuck]

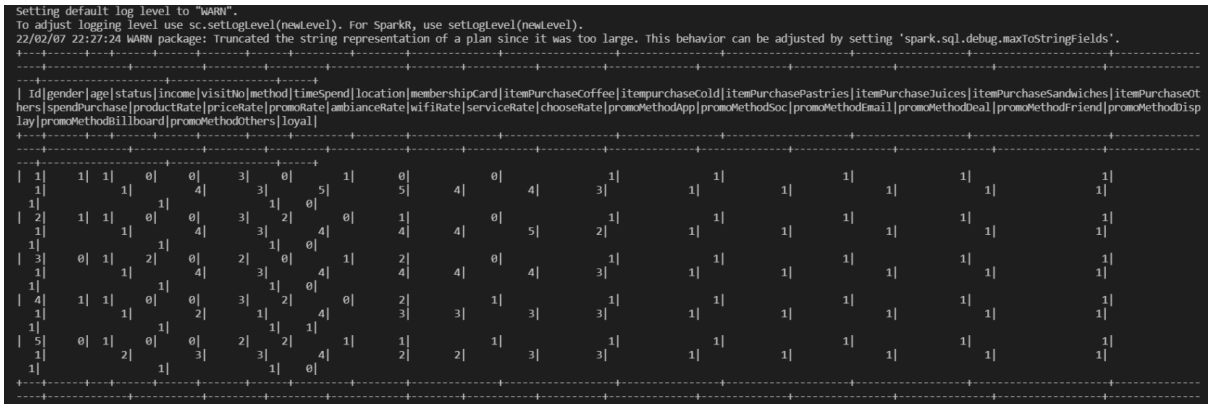
Tujuan: untuk mensurvei perilaku customer pada pelaku bisnis (starbucks).

Manfaat: Kegiatan survei customer ini dapat memprediksi perilaku customer pada suatu bisnis. sehingga perilaku usaha dapat melakukan strategi untuk kedepannya.

1. Lakukan load dan read data ke dalam format dataframe, tampilkan hasilnya.

```
# Load and read data ke dalam format dataframe
csv = spark.read.csv('dataset/customers.csv',inferSchema=True, header=True)
csv.show(5)
```

Hasil Output:



2. Lakukan pre-processing data yang meliputi:

a) Pemilihan fitur/atribut yang menunjang dari tujuan klasifikasi.

```
# Data Preprocessing - Drop Columns
dropColumns = ['itemPurchaseCoffee','itempurchaseCold','itemPurchasePastries','itemPurchaseJuices','itemPurchaseSandwiches',
               'itemPurchaseOthers','spendPurchase','productRate','priceRate','promoRate','ambianceRate','wifiRate','serviceRate',
               'chooseRate','promoMethodApp','promoMethodSoc','promoMethodEmail','promoMethodDeal','promoMethodFriend','promoMethodDisplay',
               'promoMethodBillboard','promoMethodOthers','loyal' ]
```

Pada pemilihan ini kolom data yang dihilangkan yaitu:

itemPurchaseCoffee','itempurchaseCold','itemPurchasePastries','itemPurchaseJuices' ,'itemPurchaseSandwiches','itemPurchaseOthers','spendPurchase','productRate','pric

eRate','promoRate','ambianceRate','wifiRate','serviceRate','chooseRate','promoMethodApp','promoMethodSoc','promoMethodEmail','promoMethodDeal','promoMethodFriend','promoMethodDisplay', 'promoMethodBillboard','promoMethodOthers','loyal'

b) Penanganan missing value pada data yang hilang. Jelaskan cara apa yang digunakan untuk menangani data yang hilang (dengan menghapus row, atau mengisi data yang hilang dengan metode statistik tertentu).

```
# Hasil Data setelah drop kolom
csv.show(5)
# Hasil Atribut untuk diklasifikasi
csv.printSchema()

# Data Preprocessing - Drop Duplicates
csv.dropDuplicates().show(5)

# Data Preprocessing - Handle missing data
csv.fillna('0')
csv.na.drop().show(5)
```

Dalam menangani missing value menggunakan function *fillna()* dengan nilai 0.

c) Penanganan data yang duplikasi, tampilkan data yang terduplikasi dan hasil setelah penanganan.

Data yang terduplikasi:

```
+---+------+---+------+------+-------+------+---------+--------+--------------+
| Id|gender|age|status|income|visitNo|method|timeSpend|location|membershipCard|
+---+------+---+------+------+-------+------+---------+--------+--------------+
| 40|     0|  2|     2|     2|      3|     2|        0|       0|             0|
|  3|     0|  1|     2|     0|      2|     0|        1|       2|             0|
| 71|     1|  1|     2|     0|      3|     2|        0|       0|             1|
| 85|     0|  1|     2|     1|      3|     0|        2|       2|             1|
|118|     0|  3|     1|     1|      2|     0|        2|       1|             0|
+---+------+---+------+------+-------+------+---------+--------+--------------+
only showing top 5 rows

Data yang sudah didrop duplicates : 113
```

Hasil setelah penanganan:

```
+---+------+---+------+------+-------+------+---------+--------+--------------+
| Id|gender|age|status|income|visitNo|method|timeSpend|location|membershipCard|
+---+------+---+------+------+-------+------+---------+--------+--------------+
|  1|     1|  1|     0|     0|      3|     0|        1|       0|             0|
|  2|     1|  1|     0|     0|      3|     2|        0|       1|             0|
|  3|     0|  1|     2|     0|      2|     0|        1|       2|             0|
|  4|     1|  1|     0|     0|      3|     2|        0|       2|             1|
|  5|     0|  1|     0|     0|      2|     2|        1|       1|             1|
+---+------+---+------+------+-------+------+---------+--------+--------------+
only showing top 5 rows

Data yang sudah dihandle missing data : 113
```

3. Tentukan data yang digunakan sebagai data training dan data testing, masukkan ke dalam sebuah variabel.

```
# Membagi data 70% untuk data training, 30 % untuk data testing
dividedData = csv.randomSplit([0.7, 0.3])
trainingData = dividedData[0] #index 0 = data training
testingData = dividedData[1] #index 1 = data testing
train_rows = trainingData.count()
test_rows = testingData.count()
print ("Training data rows:", train_rows, "; Testing data rows:", test_rows)
```

Membagi data training 70% dan data testing 30%.

Hasil output:

```
+---+------+---+------+------+-------+------+---------+--------+--------------+
| Id|gender|age|status|income|visitNo|method|timeSpend|location|membershipCard|
+---+------+---+------+------+-------+------+---------+--------+--------------+
|  1|     1|  1|     0|     0|      3|     0|        1|       0|             0|
|  2|     1|  1|     0|     0|      3|     2|        0|       1|             0|
|  3|     0|  1|     2|     0|      2|     0|        1|       2|             0|
|  4|     1|  1|     0|     0|      3|     2|        0|       2|             1|
|  5|     0|  1|     0|     0|      2|     2|        1|       1|             1|
+---+------+---+------+------+-------+------+---------+--------+--------------+
only showing top 5 rows

Training data rows: 83 ; Testing data rows: 30
```

4. Lakukan pemodelan dengan metode klasifikasi yang digunakan.

```
# 4 Inisialisasi klasifikasi dengan metode regresi
classifier = LogisticRegression(
    labelCol="label",featuresCol="features",maxIter=10,regParam=0.3)
# uji data klasifikasi
model = classifier.fit(trainingDataFinal)
print ("Classifier model is trained!")
```

Menggunakan metode regresi dengan function LogisticRegression().

Hasil output:

```
Classifier model is trained!
```

Melakukan pengelompokan data training hasil data test.

```
#Inisialisasi Penggabungan data
assembler = VectorAssembler(inputCols = [
    "gender", "age", "status", "income","Id",
    "timeSpend","location","membershipCard"], outputCol="features")
trainingDataFinal = assembler.transform(
    trainingData).select(col("features"), col("visitNo").alias("label"))
trainingDataFinal.show(truncate=False, n=2)
```

Hasil output:

```
+--------------------------------+-----+
|features                        |label|
+--------------------------------+-----+
|(8,[0,1,4,6],[1.0,1.0,2.0,1.0]) |3    |
|[0.0,1.0,2.0,0.0,3.0,1.0,2.0,0.0]|2    |
+--------------------------------+-----+
only showing top 2 rows
```

5. Lakukan prediksi hasil data test.

```
# 5 Prediksi data testing
prediction = model.transform(testingDataFinal)
predictionFinal = prediction.select(
    "features", "prediction", "probability", "trueLabel")
predictionFinal.show(truncate=False, n=3)
prediction.show(truncate=False, n=3)
```

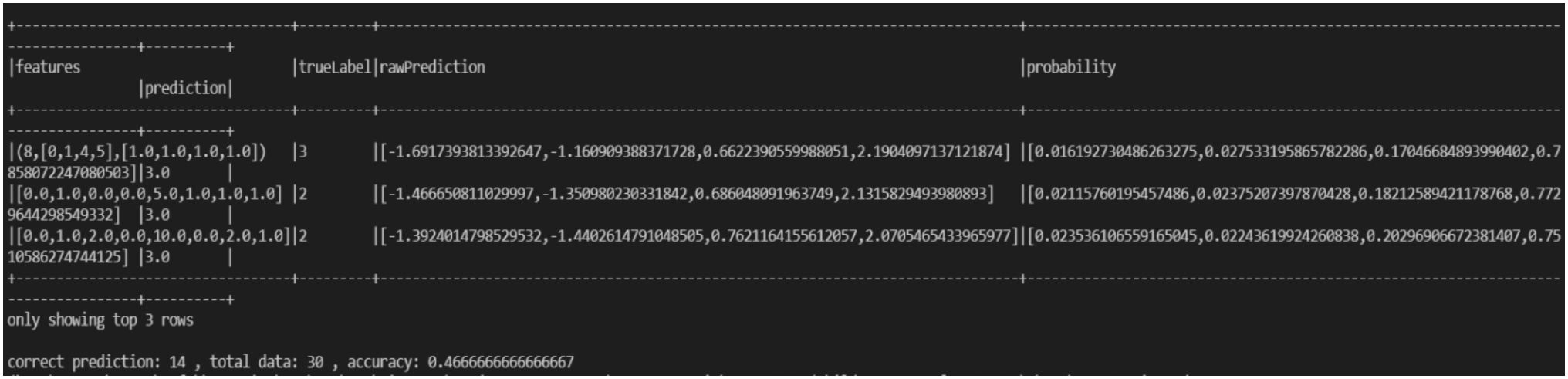
Hasil output:

```
+--------------------------------+----------+----------------------------------------------------------------------------------------+---------+
|features                        |prediction|probability                                                                             |trueLabel|
+--------------------------------+----------+----------------------------------------------------------------------------------------+---------+
|(8,[0,1,4,5],[1.0,1.0,1.0,1.0]) |3.0       |[0.016192730486263275,0.027533195865782286,0.17046684893990402,0.7858072247080503]|3        |
|[0.0,1.0,0.0,0.0,5.0,1.0,1.0,1.0]|3.0       |[0.02115760195457486,0.02375207397870428,0.18212589421178768,0.7729644298549332]  |2        |
|[0.0,1.0,2.0,0.0,10.0,0.0,2.0,1.0]|3.0       |[0.023536106559165045,0.02243619924260838,0.20296906672381407,0.7510586274744125] |2        |
+--------------------------------+----------+----------------------------------------------------------------------------------------+---------+
only showing top 3 rows
```

6. Hitung evaluasi performance dari hasil prediksi.

```
# 6 Hitung performa model
correctPrediction = predictionFinal.filter(
    predictionFinal['prediction'] == predictionFinal['trueLabel']).count()
totalData = predictionFinal.count()
print("correct prediction:", correctPrediction, ", total data:", totalData,
      ", accuracy:", correctPrediction/totalData)
```

Hasil output:



```
+--------------------------------+---------+----------------------------------------------------------------------------------------+----------------------------------------------------------------------------------------+----------+
|features                        |trueLabel|rawPrediction                                                                           |probability                                                                             |prediction|
+--------------------------------+---------+----------------------------------------------------------------------------------------+----------------------------------------------------------------------------------------+----------+
|(8,[0,1,4,5],[1.0,1.0,1.0,1.0]) |3        |[-1.6917393813392647,-1.160909388371728,0.6622390559988051,2.1904097137121874]         |[0.016192730486263275,0.027533195865782286,0.17046684893990402,0.785807224708050 3]|3.0       |
|[0.0,1.0,0.0,0.0,5.0,1.0,1.0,1.0]|2        |[-1.466650811029997,-1.350980230331842,0.686048091963749,2.1315829493980893]           |[0.02115760195457486,0.02375207397870428,0.18212589421178768,0.7729644298549332]       |3.0       |
|[0.0,1.0,2.0,0.0,10.0,0.0,2.0,1.0]|2        |[-1.3924014798529532,-1.4402614791048505,0.7621164155612057,2.070546543396597 7]|[0.023536106559165045,0.02243619924260838,0.20296906672381407,0.7510586274744125]       |3.0       |
+--------------------------------+---------+----------------------------------------------------------------------------------------+----------------------------------------------------------------------------------------+----------+
only showing top 3 rows

correct prediction: 14 , total data: 30 , accuracy: 0.4666666666666667
```