Project 2 Option-1 Part 3

P3: Logistic Regression classifier on Census Income Data

In this section, we will analyze the Census Dataset from the UCI Machine Learning Repository (link). It provides the following data,

1. Training Set: adult.data (link)
2. Test Set: adult.test (link)
3. Data Description: adult.name (link)

The data contains anonymous information such as age, occupation, education, working class, etc. The goal is to train a binary classifier to predict the income which has two possible values $> 50K$ and $< 50K$. There are 48842 instances and 14 attributes in the dataset. The data contains a good blend of categorical, numerical and missing values.

We use the dataset from Kaggle:

https://www.kaggle.com/uciml/adult-census-income

Data Preprocessing:

Before train logistic regression in Spark, we want to preprocess the dataset first.

Data shape = (32561, 15)

32561 rows and 15 columns

There are '?' in the dataset and we dropped the all the data with a '?' As shown below, even after we dropped all the null values, we did not lose a significant amount of data points. Therefore, it is okay to drop all the data with a '?'. We also dropped columns 'education' and 'fnlwgt' because we think it is unnecessary. 'education.num' can replace 'education'

Data shape = (30162, 13)

32561 rows and 13 columns

Then we encoded all the catorgical features

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()

cols=("workclass","marital.status","occupation","relationship","race","sex", "native.country")
for i in cols:
    census[i]=le.fit_transform(census[i])
census.head()
```

	age	workclass	education.num	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native.country	income
1	82	2	9	6	3	1	4	0	0	4356	18	38	<=50K
3	54	2	4	0	6	4	4	0	0	3900	40	38	<=50K
4	41	2	10	5	9	3	4	0	0	3900	40	38	<=50K
5	34	2	9	0	7	4	4	0	0	3770	45	38	<=50K
6	38	2	6	5	0	4	4	1	0	3770	40	38	<=50K

Code

Initialization: To implement a Logestic Regression in Spark, it needs to initialize the session.

```
if __name__ == "__main__":
    spark = SparkSession\
        .builder\
        .appName("AdultCensus")\
        .getOrCreate()
```

Load Data:

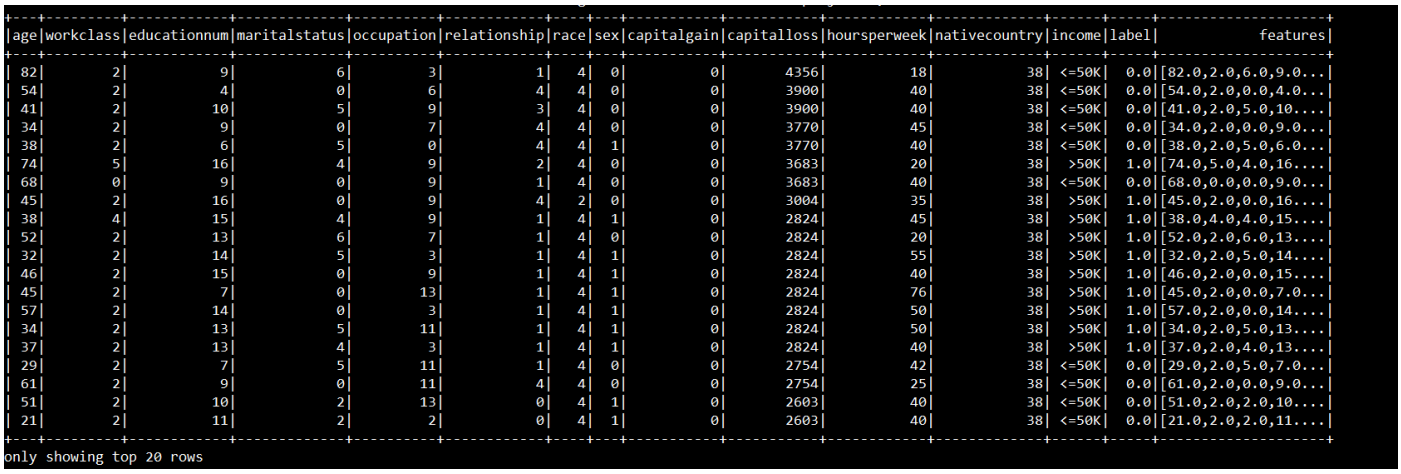
There are errors when calling the feature names, so we make sure there are no white space before or after the name. Also, we remove the "." in the features to avoid any unnesscary error.

```
18  #load data
19  dataset = spark.read.format("csv").load("/logistic_regression/input/census_clean.csv",
20                                          header="true", inferSchema="true",
21                                          ignoreLeadingWhiteSpace='true',
22                                          ignoreTrailingWhiteSpace='true')
23  #remove whitespace, replace feature name with a '.' to '' and convert to a dataframe
24  data_list = []
25  for col in dataset.columns:
26      new_name = col.strip()
27      new_name = "".join(new_name.split())
28      new_name = new_name.replace('.','')
29      data_list.append(new_name)
30  print(data_list)
31  data = dataset.toDF(*data_list)
32  #data.show()
```

```
+---+---------+------------+-------------+----------+------------+----+---+-----------+-----------+------------+-------------+------+
|age|workclass|educationnum|maritalstatus|occupation|relationship|race|sex|capitalgain|capitalloss|hoursperweek|nativecountry|income|
+---+---------+------------+-------------+----------+------------+----+---+-----------+-----------+------------+-------------+------+
| 82|        2|           9|            6|         3|           1|   4|  0|          0|       4356|          18|           38| <=50K|
| 54|        2|           4|            0|         6|           4|   4|  0|          0|       3900|          40|           38| <=50K|
| 41|        2|          10|            5|         9|           3|   4|  0|          0|       3900|          40|           38| <=50K|
| 34|        2|           9|            0|         7|           4|   4|  0|          0|       3770|          45|           38| <=50K|
| 38|        2|           6|            5|         0|           4|   4|  1|          0|       3770|          40|           38| <=50K|
| 74|        5|          16|            4|         9|           2|   4|  0|          0|       3683|          20|           38|  >50K|
| 68|        0|           9|            0|         9|           1|   4|  0|          0|       3683|          40|           38| <=50K|
| 45|        2|          16|            0|         9|           4|   2|  0|          0|       3004|          35|           38|  >50K|
| 38|        4|          15|            4|         9|           1|   4|  1|          0|       2824|          45|           38|  >50K|
| 52|        2|          13|            6|         7|           1|   4|  0|          0|       2824|          20|           38|  >50K|
| 32|        2|          14|            5|         3|           1|   4|  1|          0|       2824|          55|           38|  >50K|
| 46|        2|          15|            0|         9|           1|   4|  1|          0|       2824|          40|           38|  >50K|
| 45|        2|           7|            0|        13|           1|   4|  1|          0|       2824|          76|           38|  >50K|
| 57|        2|          14|            0|         3|           1|   4|  1|          0|       2824|          50|           38|  >50K|
| 34|        2|          13|            5|        11|           1|   4|  1|          0|       2824|          50|           38|  >50K|
| 37|        2|          13|            4|         3|           1|   4|  1|          0|       2824|          40|           38|  >50K|
| 29|        2|           7|            5|        11|           1|   4|  0|          0|       2754|          42|           38| <=50K|
| 61|        2|           9|            0|        11|           4|   4|  0|          0|       2754|          25|           38| <=50K|
| 51|        2|          10|            2|        13|           0|   4|  1|          0|       2603|          40|           38| <=50K|
| 21|        2|          11|            2|         2|           0|   4|  1|          0|       2603|          40|           38| <=50K|
+---+---------+------------+-------------+----------+------------+----+---+-----------+-----------+------------+-------------+------+
only showing top 20 rows
```

We need to vectorize the features and change binary label to index of 0 and 1.

```
34  #add a label column using feature income
35  index = StringIndexer(inputCol = 'income', outputCol = 'label')
36  data_all =index.fit(data).transform(data)
37
38  #vectorize features
39  assembler = VectorAssembler(inputCols= ['age', 'workclass', 'maritalstatus',
40                                          'educationnum', 'occupation', 'relationship',
41                                          'race', 'sex', 'capitalgain', 'capitalloss',
42                                          'hoursperweek', 'nativecountry'],
43                                         outputCol="features")
44  features = assembler.transform(data_all)
45  #features.show()
46
```



age	workclass	educationnum	maritalstatus	occupation	relationship	race	sex	capitalgain	capitalloss	hoursperweek	nativecountry	income	label	features
82	2	9	6	3	1	4	0	0	4356	18	38	<=50K	0.0	[82.0,2.0,6.0,9.0...
54	2	4	0	6	4	4	0	0	3900	40	38	<=50K	0.0	[54.0,2.0,0.0,4.0...
41	2	10	5	9	3	4	0	0	3900	40	38	<=50K	0.0	[41.0,2.0,5.0,10....
34	2	9	0	7	4	4	0	0	3770	45	38	<=50K	0.0	[34.0,2.0,0.0,9.0...
38	2	6	5	0	4	4	1	0	3770	40	38	<=50K	0.0	[38.0,2.0,5.0,6.0...
74	5	16	4	9	2	4	0	0	3683	20	38	>50K	1.0	[74.0,5.0,4.0,16....
68	0	9	0	9	1	4	0	0	3683	40	38	<=50K	0.0	[68.0,0.0,0.0,9.0...
45	2	16	0	9	4	2	0	0	3004	35	38	>50K	1.0	[45.0,2.0,0.0,16....
38	4	15	4	9	1	4	1	0	2824	45	38	>50K	1.0	[38.0,4.0,4.0,15....
52	2	13	6	7	1	4	0	0	2824	20	38	>50K	1.0	[52.0,2.0,6.0,13....
32	2	14	5	3	1	4	1	0	2824	55	38	>50K	1.0	[32.0,2.0,5.0,14....
46	2	15	0	9	1	4	1	0	2824	40	38	>50K	1.0	[46.0,2.0,0.0,15....
45	2	7	0	13	1	4	1	0	2824	76	38	>50K	1.0	[45.0,2.0,0.0,7.0...
57	2	14	0	3	1	4	1	0	2824	50	38	>50K	1.0	[57.0,2.0,0.0,14....
34	2	13	5	11	1	4	1	0	2824	50	38	>50K	1.0	[34.0,2.0,5.0,13....
37	2	13	4	3	1	4	1	0	2824	40	38	>50K	1.0	[37.0,2.0,4.0,13....
29	2	7	5	11	1	4	0	0	2754	42	38	<=50K	0.0	[29.0,2.0,5.0,7.0...
61	2	9	0	11	4	4	0	0	2754	25	38	<=50K	0.0	[61.0,2.0,0.0,9.0...
51	2	10	2	13	0	4	1	0	2603	40	38	<=50K	0.0	[51.0,2.0,2.0,10....
21	2	11	2	2	0	4	1	0	2603	40	38	<=50K	0.0	[21.0,2.0,2.0,11....

only showing top 20 rows

Train the Model:

Before training the model, we need to split the dataset randomly into training and testing set to 80% and 20% accordingly. First fit the model using the training set, then predict it with the testing set.

```
47      #Split data to train and test set
48      train, test = features.randomSplit((0.8, 0.2), seed=0)
49
50      #Logistic Regression Model
51      logistic = LogisticRegression(featuresCol = 'features', labelCol = 'label', maxIter = 10)
52
53      # fitting the model
54      model = logistic.fit(train)
55
56      #print the coefficients and intercept for logistic regression
57      print("Coefficients: " + str(model.coefficients))
58      print("Intercept: " + str(model.intercept))
59
60      #predict test set
61      prediction = model.transform(test)
62      #prediction.show()
63
64      # get testing accuracy
65      evaluator = BinaryClassificationEvaluator()
66      accuracy = evaluator.evaluate(prediction)
67      print(accuracy)
```

***Find full example code in the zip folder.*

Result

Coefficients: (in the order of below features)

[0.01720, -0.2478, -0.3845, 0.2902, -0.0163, -0.3299, -0.1742, 0.3774, 0.0001, 0.0007, 0.0101, -0.0496]

['age', 'workclass', 'maritalstatus', 'educationnum', 'occupation', 'relationship', 'race', 'sex', 'capitalgain', 'capitalloss', 'hoursperweek', 'nativecountry']

Xingrou Mei

Based on the coefficients, we can see that gender and education level have a high impact on income. Features that have less impact on income are marital status and relationship.

Intercept: -1.3710

```
2021-05-12 02:23:53 INFO  ContextCleaner:54 - Cleaned accumulator 347
2021-05-12 02:23:53 INFO  BlockManagerInfo:54 - Removed broadcast_14_piece0 on node-1.c.noted-field-305223.internal:41655 in memory (size: 20.8 KB, free: 366.2 MB)
Coefficients: [0.0171990403335,-0.247811700631,-0.384541013981,0.290202221502,-0.0162710255387,-0.319881462409,-0.174211985333,0.377368621773,0.000142550486195,0.000675970797775,0.010126791
9299,-0.0495811858575]
2021-05-12 02:23:53 INFO  BlockManagerInfo:54 - Removed broadcast_14_piece0 on 10.128.0.4:40019 in memory (size: 20.8 KB, free: 366.2 MB)
Intercept: -1.37102169067
2021-05-12 02:23:53 INFO  ContextCleaner:54 - Cleaned accumulator 378
2021-05-12 02:23:53 INFO  ContextCleaner:54 - Cleaned accumulator 355
2021-05-12 02:23:53 INFO  ContextCleaner:54 - Cleaned accumulator 193
```

Testing Accuracy: 0.8199, 81.99%

```
2021-05-12 02:28:10 INFO  TaskSchedulerImpl:54 - Removed TaskSet 25.0, whose tasks have all completed, from pool
2021-05-12 02:28:10 INFO  DAGScheduler:54 - ResultStage 25 (aggregate at AreaUnderCurve.scala:45) finished in 0.098 s
2021-05-12 02:28:10 INFO  DAGScheduler:54 - Job 18 finished: aggregate at AreaUnderCurve.scala:45, took 0.102241 s
2021-05-12 02:28:10 INFO  MapPartitionsRDD:54 - Removing RDD 58 from persistence list
2021-05-12 02:28:10 INFO  BlockManager:54 - Removing RDD 58
0.819855857461
2021-05-12 02:28:10 INFO  SparkContext:54 - Invoking stop() from shutdown hook
2021-05-12 02:28:10 INFO  AbstractConnector:318 - Stopped Spark@3020bda3{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2021-05-12 02:28:10 INFO  SparkUI:54 - Stopped Spark web UI at http://node-1.c.noted-field-305223.internal:4040
2021-05-12 02:28:10 INFO  StandaloneSchedulerBackend:54 - Shutting down all executors
2021-05-12 02:28:10 INFO  CoarseGrainedSchedulerBackend$DriverEndpoint:54 - Asking each executor to shut down
```

Test.sh

```
root@node-1:/spark-examples/test-mllib/python/logistic_regression# cat test.sh
#!/bin/bash
source ../../../env.sh
/usr/local/hadoop/bin/hdfs dfs -rm -r /logistic_regression/input/
/usr/local/hadoop/bin/hdfs dfs -mkdir -p /logistic_regression/input/
/usr/local/hadoop/bin/hdfs dfs -copyFromLocal ../../../test-data/census_clean.csv /logistic_regression/input/
/usr/local/spark/bin/spark-submit --master=spark://$SPARK_MASTER:7077 ./Q3.py
```