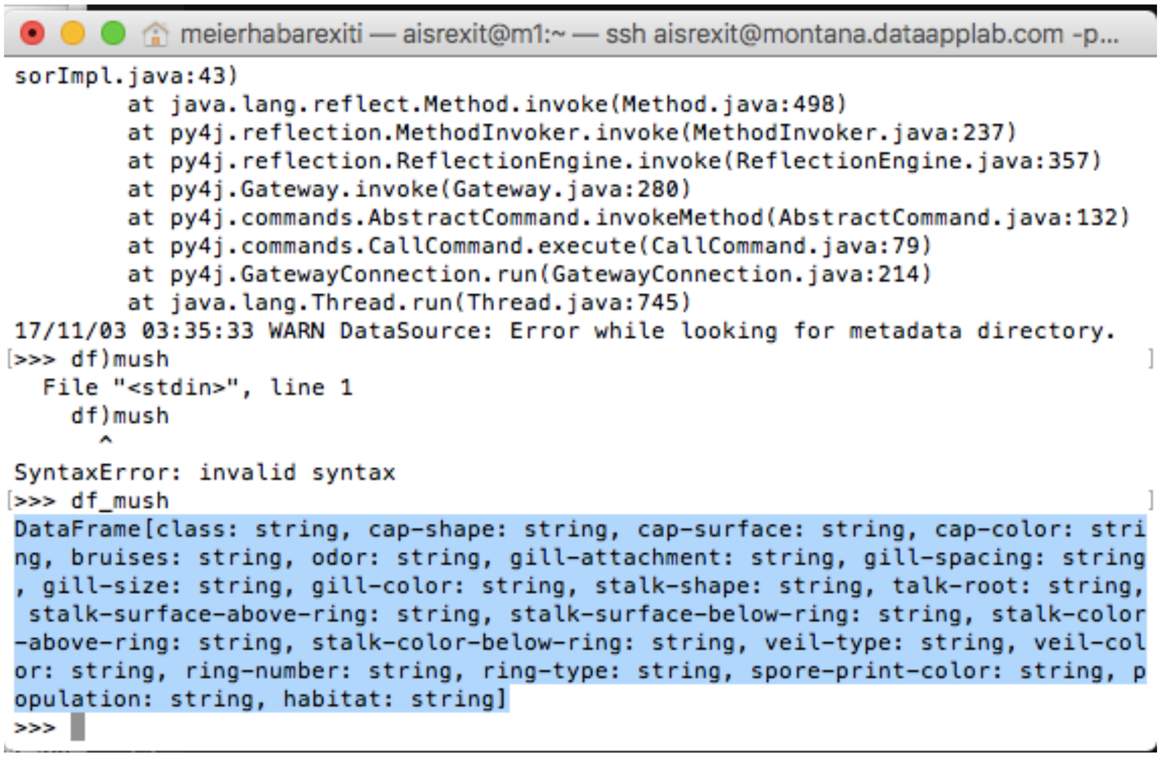
Meierhaba Rexiti
Mini Project 7 11/03/2017

```
######## Mini 7
#################### part 1: load data into HDFS
# log onto server
ssh aisrexit@montana.dataapplab.com -p 49233
mkdir mushroom # create a file mushroom in server
cd mushroom

# copy local csv file to file mushroom in server  (from a new terminal window)
scp -P 49233 ./mushrooms.csv aisrexit@montana
dataapplab.com:/home/aisrexit/mushroom

# upload the file into an existed file named 'data' in HDFS
hdfs dfs -put mushrooms.csv hdfs:///user/aisrexit/data

# double check if data is successfully uploaded to 'data'
hdfs dfs -ls hdfs:///user/aisrexit/data

#####
#################### part 2: Pyspark
SPARK_MAJOR_VERSION=2 pyspark
from pyspark.sql import SQLContext
from pyspark.sql.types import *
from pyspark.ml import Pipeline
from pyspark.ml.classification import DecisionTreeClassifier
from pyspark.ml.feature import StringIndexer, VectorIndexer, VectorAssembler
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Create a structure
struct = StructType([
        StructField('class',StringType(),True),
        StructField('cap-shape',StringType(),True),
        StructField('cap-surface',StringType(),True),
        StructField('cap-color',StringType(),True),
        StructField('bruises',StringType(),True),
        StructField('odor',StringType(),True),
        StructField('gill-attachment',StringType(),True),
        StructField('gill-spacing',StringType(),True),
        StructField('gill-size',StringType(),True),
        StructField('gill-color',StringType(),True),
        StructField('stalk-shape',StringType(),True),
        StructField('talk-root',StringType(),True),
        StructField('stalk-surface-above-ring',StringType(),True),
        StructField('stalk-surface-below-ring',StringType(),True),
        StructField('stalk-color-above-ring',StringType(),True),
        StructField('stalk-color-below-ring',StringType(),True),
        StructField('veil-type',StringType(),True),
        StructField('veil-color',StringType(),True),
        StructField('ring-number',StringType(),True),
        StructField('ring-type',StringType(),True),
        StructField('spore-print-color',StringType(),True),
        StructField('population',StringType(),True),
        StructField('habitat',StringType(),True),
       ])
```

# find the path for mushroom.csv
hdfs dfs -ls /user/aisrexit/data/

# load data as DF
df_mush =
spark.read.csv('hdfs:///user/aisrexit/data/mushrooms.csv',header=True,schema=struct)

df_mush



```
sorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:237)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:357)
        at py4j.Gateway.invoke(Gateway.java:280)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:214)
        at java.lang.Thread.run(Thread.java:745)
17/11/03 03:35:33 WARN DataSource: Error while looking for metadata directory.
>>> df)mush
  File "<stdin>", line 1
    df)mush
      ^
SyntaxError: invalid syntax
>>> df_mush
DataFrame[class: string, cap-shape: string, cap-surface: string, cap-color: stri
ng, bruises: string, odor: string, gill-attachment: string, gill-spacing: string
, gill-size: string, gill-color: string, stalk-shape: string, talk-root: string,
 stalk-surface-above-ring: string, stalk-surface-below-ring: string, stalk-color
-above-ring: string, stalk-color-below-ring: string, veil-type: string, veil-col
or: string, ring-number: string, ring-type: string, spore-print-color: string, p
opulation: string, habitat: string]
>>> 
```

#### Indexing
## features
# index features : good
indexers = [StringIndexer(inputCol=column, outputCol=column+"_index").fit(df_mush) for column in list(set(df_mush.columns))]

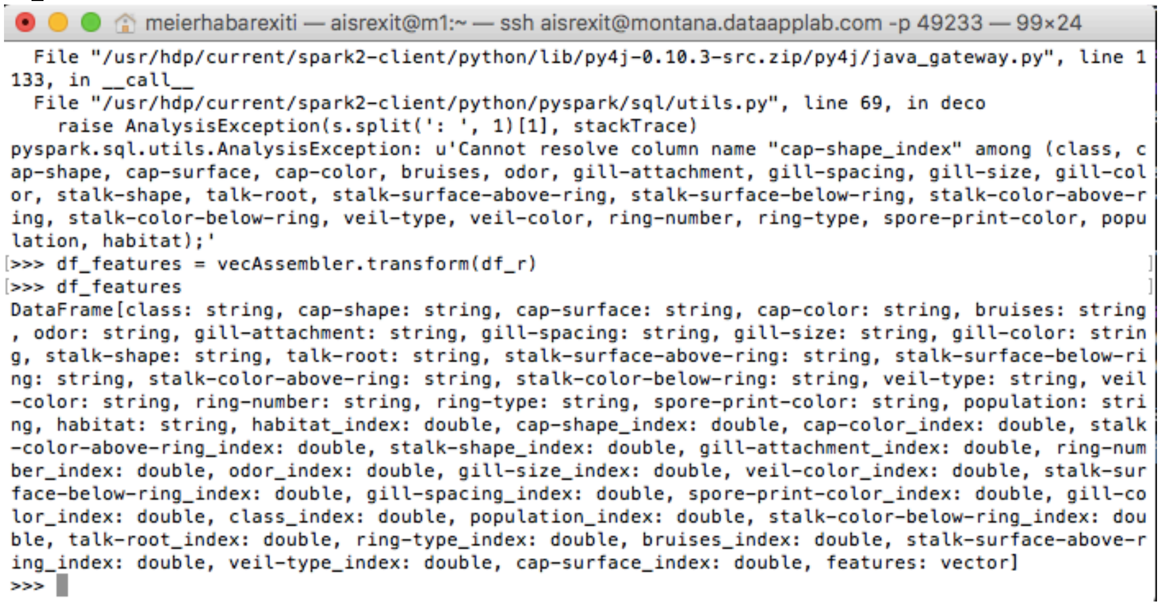
pipeline = Pipeline(stages=indexers)

df_r = pipeline.fit(df_mush).transform(df_mush)

df_r.show() # the new dataframe with coloumn and columm+index

# Assemble all features: all features in VectorAssembler has to be numeric
vecAssembler = VectorAssembler(inputCols=["cap-shape_index","cap-surface_index","cap-color_index","bruises_index","odor_index","gill-attachment_index","gill-spacing_index","gill-size_index","gill-color_index","stalk-shape_index","talk-root_index","stalk-surface-above-ring_index","stalk-surface-below-ring_index","stalk-color-above-ring_index","stalk-color-below-ring_index","veil-type_index","veil-color_index","ring-number_index","ring-type_index","spore-print-color_index","population_index","habitat_index"],outputCol="features")
vecAssembler

df_features = vecAssembler.transform(df_r)
df_features:

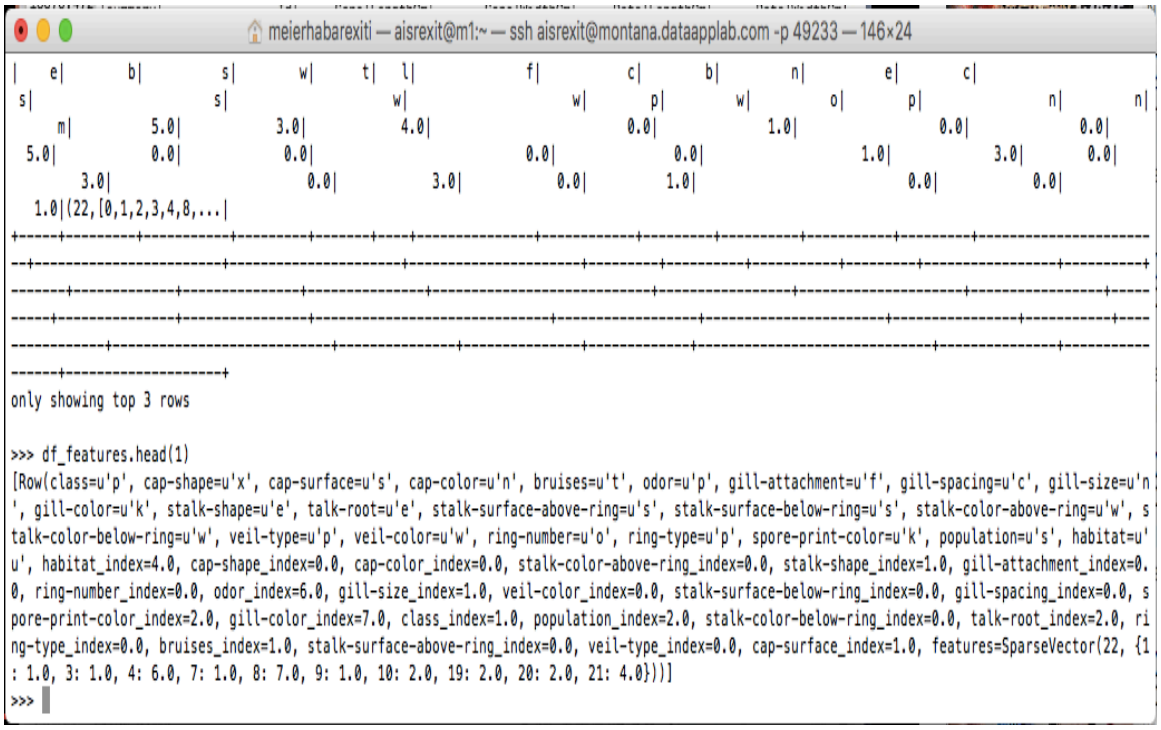


meierhabarexiti — aisrexit@m1:~ — ssh aisrexit@montana.dataapplab.com -p 49233 — 99×24

```
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.3-src.zip/py4j/java_gateway.py", line 1
133, in __call__
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/utils.py", line 69, in deco
    raise AnalysisException(s.split(': ', 1)[1], stackTrace)
pyspark.sql.utils.AnalysisException: u'Cannot resolve column name "cap-shape_index" among (class, c
ap-shape, cap-surface, cap-color, bruises, odor, gill-attachment, gill-spacing, gill-size, gill-col
or, stalk-shape, talk-root, stalk-surface-above-ring, stalk-surface-below-ring, stalk-color-above-r
ing, stalk-color-below-ring, veil-type, veil-color, ring-number, ring-type, spore-print-color, popu
lation, habitat);'
>>> df_features = vecAssembler.transform(df_r)
>>> df_features
DataFrame[class: string, cap-shape: string, cap-surface: string, cap-color: string, bruises: string
, odor: string, gill-attachment: string, gill-spacing: string, gill-size: string, gill-color: strin
g, stalk-shape: string, talk-root: string, stalk-surface-above-ring: string, stalk-surface-below-ri
ng: string, stalk-color-above-ring: string, stalk-color-below-ring: string, veil-type: string, veil
-color: string, ring-number: string, ring-type: string, spore-print-color: string, population: stri
ng, habitat: string, habitat_index: double, cap-shape_index: double, cap-color_index: double, stalk
-color-above-ring_index: double, stalk-shape_index: double, gill-attachment_index: double, ring-num
ber_index: double, odor_index: double, gill-size_index: double, veil-color_index: double, stalk-sur
face-below-ring_index: double, gill-spacing_index: double, spore-print-color_index: double, gill-co
lor_index: double, class_index: double, population_index: double, stalk-color-below-ring_index: dou
ble, talk-root_index: double, ring-type_index: double, bruises_index: double, stalk-surface-above-r
ing_index: double, veil-type_index: double, cap-surface_index: double, features: vector]
>>> 
```

df_features.head(1)

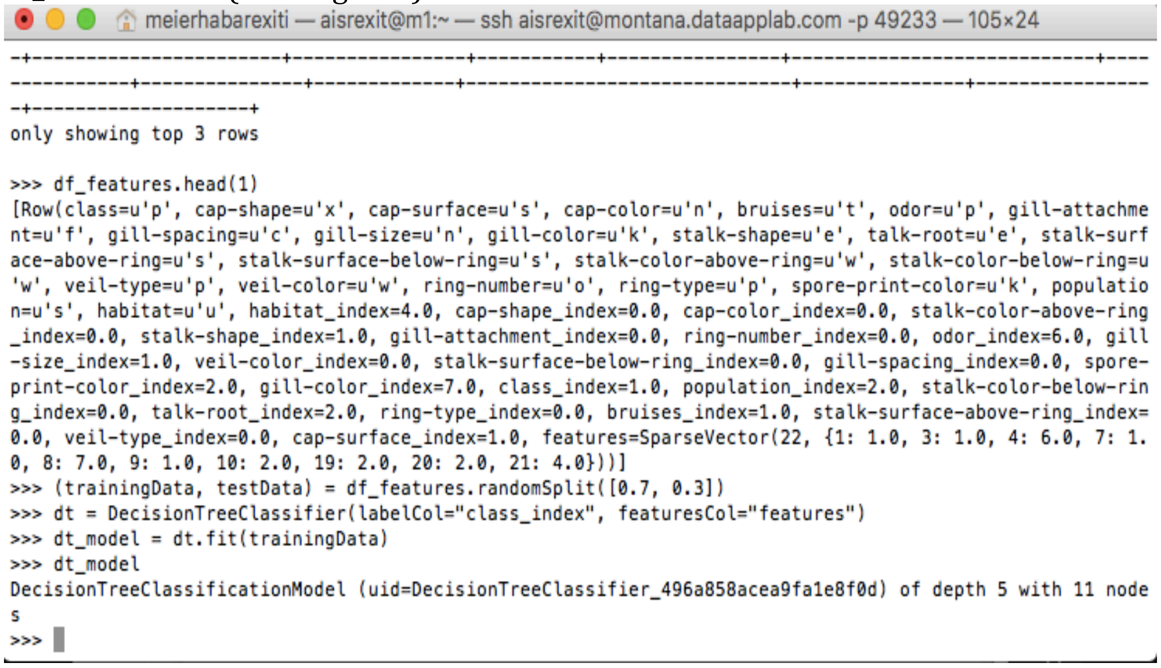
Meierhaba Rexiti
Mini Project 7 11/03/2017



```
|    e|       b|         s|       w|     t|  l|            f|        c|       b|        n|         e|       c|
 s|                     s|                  w|                 w|       p|         w|        o|        p|              n|        n|
      m|         5.0|          3.0|          4.0|                       0.0|           1.0|            0.0|          0.0|
 5.0|         0.0|          0.0|                      0.0|          0.0|                  1.0|         3.0|      0.0|
        3.0|                    0.0|          3.0|          0.0|       1.0|                       0.0|          0.0|
  1.0|(22,[0,1,2,3,4,8,...|
+-----+---------+---------+--------+----+-------------+-----------+--------+---------+---------+--------+-----------
--+-----------------------+-------------------+-------------------+--------+---------+--------+--------------+---------+
-------+-----------+--------------+-----------------------+---------------+-----------------+-------------+-----
----+-------------+--------------+-------------------------+---------------+-------------------+-------------+----
-----------+------------------------+-------------+--------------+---------------+--------------------+-------------
------+-------------------+
only showing top 3 rows

>>> df_features.head(1)
[Row(class=u'p', cap-shape=u'x', cap-surface=u's', cap-color=u'n', bruises=u't', odor=u'p', gill-attachment=u'f', gill-spacing=u'c', gill-size=u'n
', gill-color=u'k', stalk-shape=u'e', talk-root=u'e', stalk-surface-above-ring=u's', stalk-surface-below-ring=u's', stalk-color-above-ring=u'w', s
talk-color-below-ring=u'w', veil-type=u'p', veil-color=u'w', ring-number=u'o', ring-type=u'p', spore-print-color=u'k', population=u's', habitat=u'
u', habitat_index=4.0, cap-shape_index=0.0, cap-color_index=0.0, stalk-color-above-ring_index=0.0, stalk-shape_index=1.0, gill-attachment_index=0.
0, ring-number_index=0.0, odor_index=6.0, gill-size_index=1.0, veil-color_index=0.0, stalk-surface-below-ring_index=0.0, gill-spacing_index=0.0, s
pore-print-color_index=2.0, gill-color_index=7.0, class_index=1.0, population_index=2.0, stalk-color-below-ring_index=0.0, talk-root_index=2.0, ri
ng-type_index=0.0, bruises_index=1.0, stalk-surface-above-ring_index=0.0, veil-type_index=0.0, cap-surface_index=1.0, features=SparseVector(22, {1
: 1.0, 3: 1.0, 4: 6.0, 7: 1.0, 8: 7.0, 9: 1.0, 10: 2.0, 19: 2.0, 20: 2.0, 21: 4.0}))]
>>>
```

### Fitting model
## train test split
(trainingData, testData) = df_features.randomSplit([0.7, 0.3])

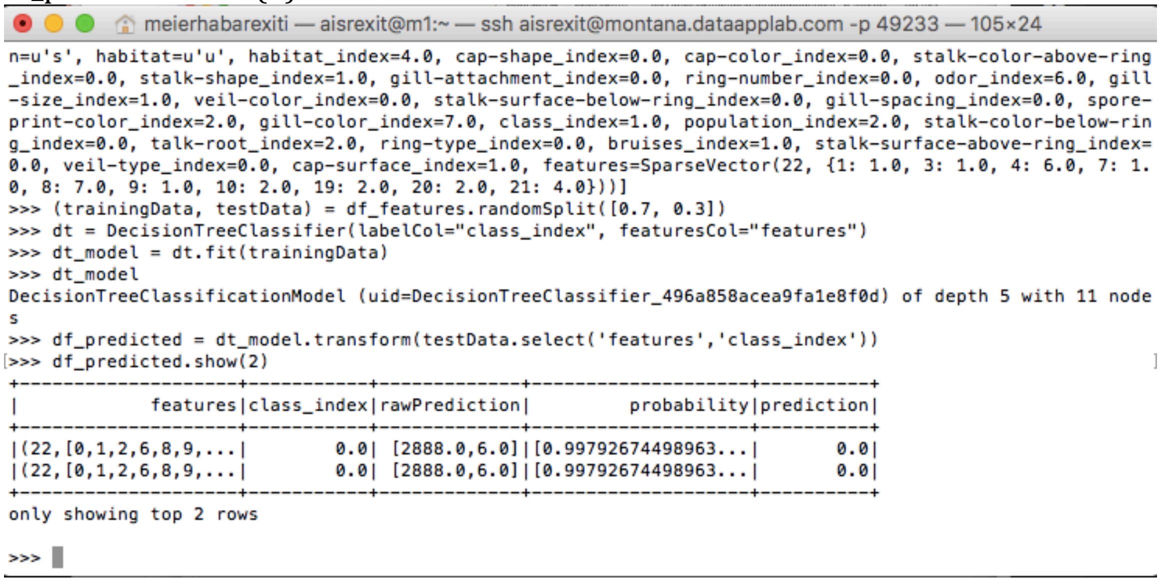
## fitting a decision tree model
dt = DecisionTreeClassifier(labelCol="class_index", featuresCol="features")
dt_model = dt.fit(trainingData)



```
-+-----------------------+----------------+-----------+----------------+----------------------------+----
-----------+---------------+-------------+------------------------------+---------------+---------------
-+--------------------+
only showing top 3 rows

>>> df_features.head(1)
[Row(class=u'p', cap-shape=u'x', cap-surface=u's', cap-color=u'n', bruises=u't', odor=u'p', gill-attachme
nt=u'f', gill-spacing=u'c', gill-size=u'n', gill-color=u'k', stalk-shape=u'e', talk-root=u'e', stalk-surf
ace-above-ring=u's', stalk-surface-below-ring=u's', stalk-color-above-ring=u'w', stalk-color-below-ring=u
'w', veil-type=u'p', veil-color=u'w', ring-number=u'o', ring-type=u'p', spore-print-color=u'k', populatio
n=u's', habitat=u'u', habitat_index=4.0, cap-shape_index=0.0, cap-color_index=0.0, stalk-color-above-ring
_index=0.0, stalk-shape_index=1.0, gill-attachment_index=0.0, ring-number_index=0.0, odor_index=6.0, gill
-size_index=1.0, veil-color_index=0.0, stalk-surface-below-ring_index=0.0, gill-spacing_index=0.0, spore-
print-color_index=2.0, gill-color_index=7.0, class_index=1.0, population_index=2.0, stalk-color-below-rin
g_index=0.0, talk-root_index=2.0, ring-type_index=0.0, bruises_index=1.0, stalk-surface-above-ring_index=
0.0, veil-type_index=0.0, cap-surface_index=1.0, features=SparseVector(22, {1: 1.0, 3: 1.0, 4: 6.0, 7: 1.
0, 8: 7.0, 9: 1.0, 10: 2.0, 19: 2.0, 20: 2.0, 21: 4.0}))]
>>> (trainingData, testData) = df_features.randomSplit([0.7, 0.3])
>>> dt = DecisionTreeClassifier(labelCol="class_index", featuresCol="features")
>>> dt_model = dt.fit(trainingData)
>>> dt_model
DecisionTreeClassificationModel (uid=DecisionTreeClassifier_496a858acea9fa1e8f0d) of depth 5 with 11 node
s
>>>
```

Meierhaba Rexiti
Mini Project 7 11/03/2017

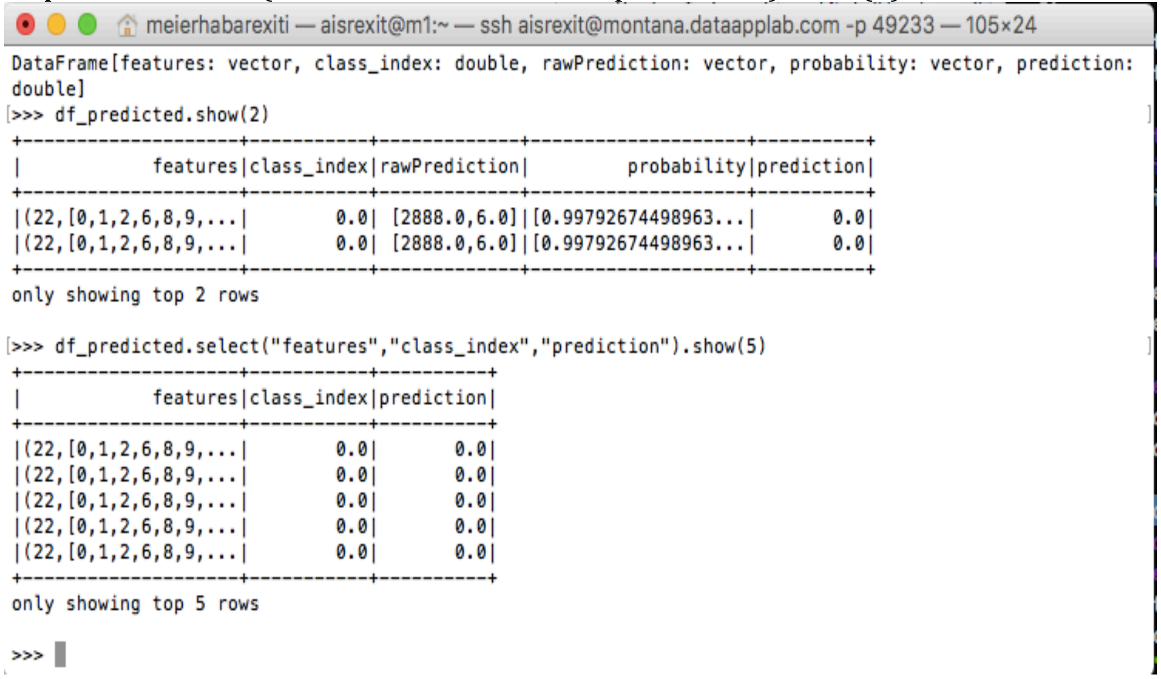
## make predictions on testdata
df_predicted = dt_model.transform(testData.select('features','class_index'))
df_predicted.show(2)



meierhabarexiti — aisrexit@m1:~ — ssh aisrexit@montana.dataapplab.com -p 49233 — 105×24

```
n=u's', habitat=u'u', habitat_index=4.0, cap-shape_index=0.0, cap-color_index=0.0, stalk-color-above-ring
_index=0.0, stalk-shape_index=1.0, gill-attachment_index=0.0, ring-number_index=0.0, odor_index=6.0, gill
-size_index=1.0, veil-color_index=0.0, stalk-surface-below-ring_index=0.0, gill-spacing_index=0.0, spore-
print-color_index=2.0, gill-color_index=7.0, class_index=1.0, population_index=2.0, stalk-color-below-rin
g_index=0.0, talk-root_index=2.0, ring-type_index=0.0, bruises_index=1.0, stalk-surface-above-ring_index=
0.0, veil-type_index=0.0, cap-surface_index=1.0, features=SparseVector(22, {1: 1.0, 3: 1.0, 4: 6.0, 7: 1.
0, 8: 7.0, 9: 1.0, 10: 2.0, 19: 2.0, 20: 2.0, 21: 4.0}))]
>>> (trainingData, testData) = df_features.randomSplit([0.7, 0.3])
>>> dt = DecisionTreeClassifier(labelCol="class_index", featuresCol="features")
>>> dt_model = dt.fit(trainingData)
>>> dt_model
DecisionTreeClassificationModel (uid=DecisionTreeClassifier_496a858acea9fa1e8f0d) of depth 5 with 11 node
s
>>> df_predicted = dt_model.transform(testData.select('features','class_index'))
>>> df_predicted.show(2)
+--------------------+-----------+-------------+--------------------+----------+
|            features|class_index|rawPrediction|         probability|prediction|
+--------------------+-----------+-------------+--------------------+----------+
|(22,[0,1,2,6,8,9,...|        0.0| [2888.0,6.0]|[0.99792674498963...|       0.0|
|(22,[0,1,2,6,8,9,...|        0.0| [2888.0,6.0]|[0.99792674498963...|       0.0|
+--------------------+-----------+-------------+--------------------+----------+
only showing top 2 rows

>>> 
```

df_predicted.select("features","class_index","prediction").show(5)



meierhabarexiti — aisrexit@m1:~ — ssh aisrexit@montana.dataapplab.com -p 49233 — 105×24

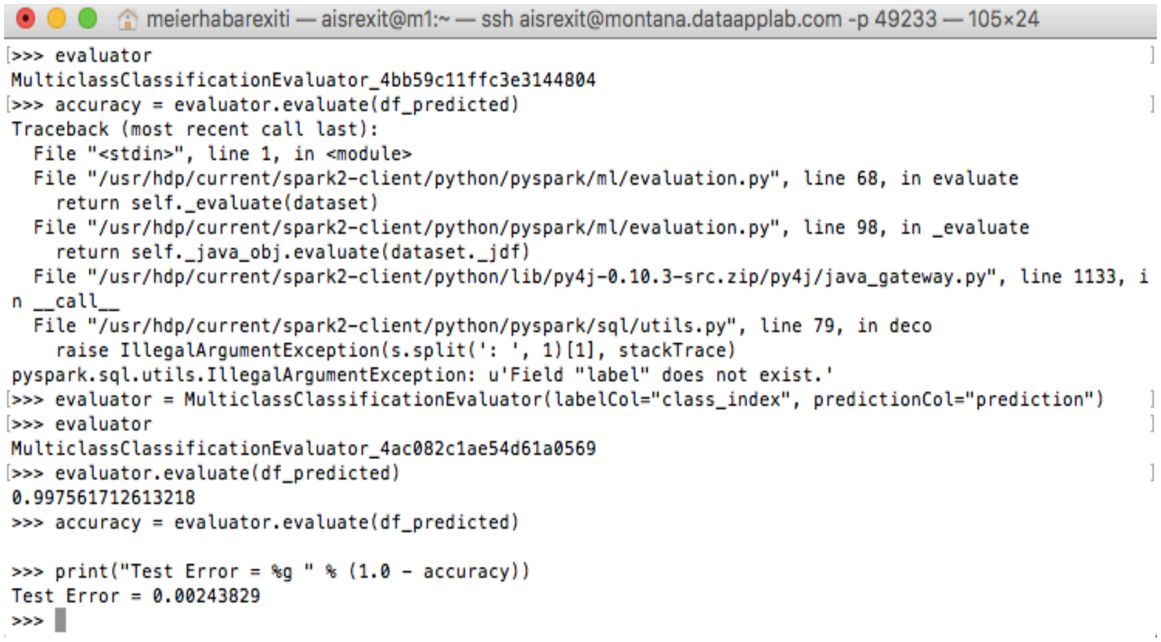
```
DataFrame[features: vector, class_index: double, rawPrediction: vector, probability: vector, prediction:
double]
>>> df_predicted.show(2)
+--------------------+-----------+-------------+--------------------+----------+
|            features|class_index|rawPrediction|         probability|prediction|
+--------------------+-----------+-------------+--------------------+----------+
|(22,[0,1,2,6,8,9,...|        0.0| [2888.0,6.0]|[0.99792674498963...|       0.0|
|(22,[0,1,2,6,8,9,...|        0.0| [2888.0,6.0]|[0.99792674498963...|       0.0|
+--------------------+-----------+-------------+--------------------+----------+
only showing top 2 rows

>>> df_predicted.select("features","class_index","prediction").show(5)
+--------------------+-----------+----------+
|            features|class_index|prediction|
+--------------------+-----------+----------+
|(22,[0,1,2,6,8,9,...|        0.0|       0.0|
|(22,[0,1,2,6,8,9,...|        0.0|       0.0|
|(22,[0,1,2,6,8,9,...|        0.0|       0.0|
|(22,[0,1,2,6,8,9,...|        0.0|       0.0|
|(22,[0,1,2,6,8,9,...|        0.0|       0.0|
+--------------------+-----------+----------+
only showing top 5 rows

>>> 
```

### Evaluation
## Select (prediction, true label) and compute test error
evaluator = MulticlassClassificationEvaluator(labelCol="class_index", predictionCol="prediction")

Meierhaba Rexiti
Mini Project 7 11/03/2017

evaluator.evaluate(df_predicted)

accuracy = evaluator.evaluate(df_predicted)
print("Test Error = %g " % (1.0 - accuracy))


meierhabarexiti — aisrexit@m1:~ — ssh aisrexit@montana.dataapplab.com -p 49233 — 105×24

```
>>> evaluator
MulticlassClassificationEvaluator_4bb59c11ffc3e3144804
>>> accuracy = evaluator.evaluate(df_predicted)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/hdp/current/spark2-client/python/pyspark/ml/evaluation.py", line 68, in evaluate
    return self._evaluate(dataset)
  File "/usr/hdp/current/spark2-client/python/pyspark/ml/evaluation.py", line 98, in _evaluate
    return self._java_obj.evaluate(dataset._jdf)
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.3-src.zip/py4j/java_gateway.py", line 1133, i
n __call__
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/utils.py", line 79, in deco
    raise IllegalArgumentException(s.split(': ', 1)[1], stackTrace)
pyspark.sql.utils.IllegalArgumentException: u'Field "label" does not exist.'
>>> evaluator = MulticlassClassificationEvaluator(labelCol="class_index", predictionCol="prediction")
>>> evaluator
MulticlassClassificationEvaluator_4ac082c1ae54d61a0569
>>> evaluator.evaluate(df_predicted)
0.997561712613218
>>> accuracy = evaluator.evaluate(df_predicted)

>>> print("Test Error = %g " % (1.0 - accuracy))
Test Error = 0.00243829
>>> 
```