```
import pyspark
from pyspark.sql import SparkSession
from pyspark.ml.regression import LinearRegression
from pyspark.ml.feature import VectorAssembler
import matplotlib.pyplot as plt
import seaborn as sns


from pylab import *
from pyspark.sql.functions import udf, concat, col, lit
from pyspark import SparkConf, SparkContext
from pyspark.sql import SparkSession, SQLContext

from pyspark.sql.types import *
import pyspark.sql.functions as F
sc = SparkContext.getOrCreate(SparkConf().setMaster("local[*]"))
from pyspark.sql import SparkSession
spark = SparkSession \
    .builder \
    .getOrCreate()
sqlContext = SQLContext(sc)
```

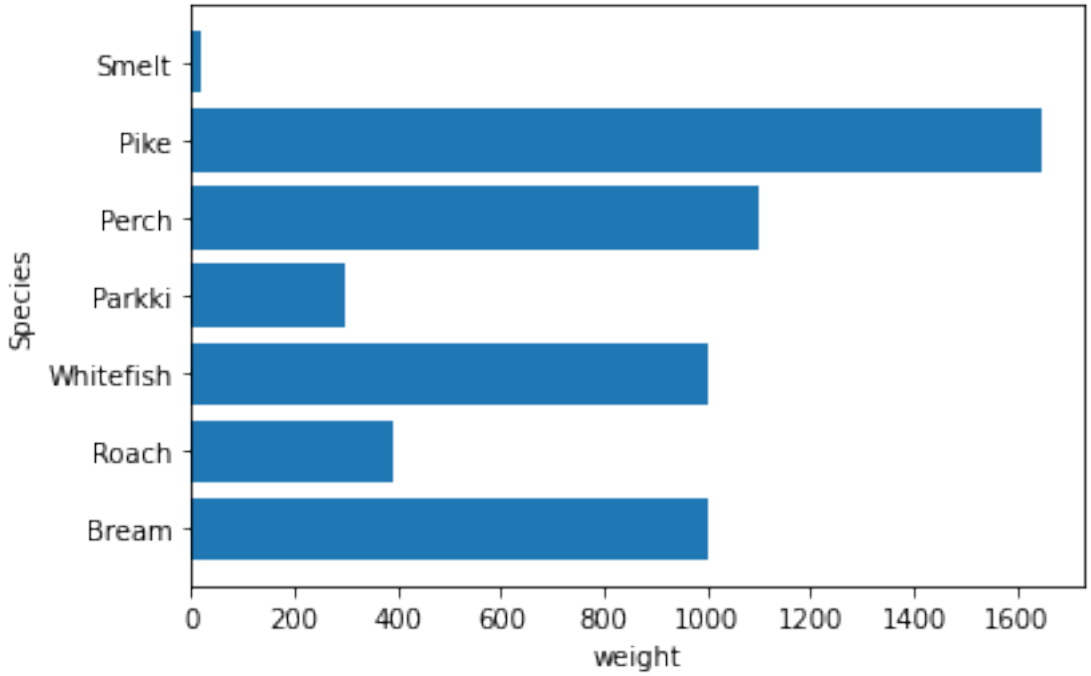
```
/databricks/spark/python/pyspark/sql/context.py:117: FutureWarning:
Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
  warnings.warn(
```

```
spark = SparkSession.builder.appName("fishes").getOrCreate()

dataset = spark.read.csv("/FileStore/tables/Fish-1.csv",
inferSchema=True, header =True)
dataset.printSchema()
```

```
root
 |-- Species: string (nullable = true)
 |-- Weight: double (nullable = true)
 |-- Length1: double (nullable = true)
 |-- Length2: double (nullable = true)
 |-- Length3: double (nullable = true)
 |-- Height: double (nullable = true)
 |-- Width: double (nullable = true)
```

```
import pandas as pd
pd.DataFrame(dataset.take(5), columns=dataset.columns).transpose()

numeric_features = [t[0] for t in dataset.dtypes if t[1] == 'double']
dataset.select(numeric_features).describe().toPandas().transpose()

dataset.createOrReplaceTempView("df_sql")
fish = spark.sql("Select Species from df_sql")
fish = fish.rdd.map(lambda row : row.Species).collect()
```

```
pre = spark.sql("Select Weight from df_sql")
pre = pre.rdd.map(lambda row : row.Weight).collect()
plt.barh(fish,pre)

plt.xlabel("weight")
plt.ylabel("Species")


plt.show()
```




```
from pyspark.ml.feature import VectorAssembler

numericCols = ['Length1','Length2','Length3','Height','Width']
assembler = VectorAssembler(inputCols=numericCols,
outputCol="features")
df = assembler.transform(dataset)
df.show()
```

```
+-------+------+-------+-------+-------+-------+------
+--------------------+
|Species|Weight|Length1|Length2|Length3| Height| Width|
features|
+-------+------+-------+-------+-------+-------+------
+--------------------+
|  Bream| 242.0|   23.2|   25.4|   30.0|  11.52|  4.02|
[23.2,25.4,30.0,1...|
|  Bream| 290.0|   24.0|   26.3|   31.2|  12.48|4.3056|
[24.0,26.3,31.2,1...|
|  Bream| 340.0|   23.9|   26.5|   31.1|12.3778|4.6961|
```

```
[23.9,26.5,31.1,1...|
|  Bream| 363.0|   26.3|   29.0|   33.5|  12.73|4.4555|
[26.3,29.0,33.5,1...|
|  Bream| 430.0|   26.5|   29.0|   34.0| 12.444| 5.134|
[26.5,29.0,34.0,1...|
|  Bream| 450.0|   26.8|   29.7|   34.7|13.6024|4.9274|
[26.8,29.7,34.7,1...|
|  Bream| 500.0|   26.8|   29.7|   34.5|14.1795|5.2785|
[26.8,29.7,34.5,1...|
|  Bream| 390.0|   27.6|   30.0|   35.0|  12.67|  4.69|
[27.6,30.0,35.0,1...|
|  Bream| 450.0|   27.6|   30.0|   35.1|14.0049|4.8438|
[27.6,30.0,35.1,1...|
|  Bream| 500.0|   28.5|   30.7|   36.2|14.2266|4.9594|
[28.5,30.7,36.2,1...|
|  Bream| 475.0|   28.4|   31.0|   36.2|14.2628|5.1042|
[28.4,31.0,36.2,1...|
|  Bream| 500.0|   28.7|   31.0|   36.2|14.3714|4.8146|
[28.7,31.0,36.2,1...|
|  Bream| 500.0|   29.1|   31.5|   36.4|13.7592| 4.368|
[29.1,31.5,36.4,1...|
|  Bream| 340.0|   29.5|   32.0|   37.3|13.9129|5.0728|
[29.5,32.0,37.3,1...|
|  Bream| 600.0|   29.4|   32.0|   37.2|14.9544|5.1708|
[29.4,32.0,37.2,1...|
|  Bream| 600.0|   29.4|   32.0|   37.2| 15.438|  5.58|
[29.4,32.0,37.2,1...|
|  Bream| 700.0|   30.4|   33.0|   38.3|14.8604|5.2854|
[30.4,33.0,38.3,1...|
|  Bream| 700.0|   30.4|   33.0|   38.5| 14.938|5.1975|
[30.4,33.0,38.5,1...|
|  Bream| 610.0|   30.9|   33.5|   38.6| 15.633|5.1338|
[30.9,33.5,38.6,1...|
|  Bream| 650.0|   31.0|   33.5|   38.7|14.4738|5.7276|
[31.0,33.5,38.7,1...|
+-------+------+-------+-------+-------+-------+------
+--------------------+
only showing top 20 rows
```

```
from pyspark.ml.feature import StringIndexer

label_stringIdx = StringIndexer(inputCol = 'Species', outputCol =
'labelIndex')
df = label_stringIdx.fit(df).transform(df)
df.show()
```

```
+-------+------+-------+-------+-------+-------+------
+--------------------+----------+
|Species|Weight|Length1|Length2|Length3| Height| Width|
```

```
features|labelIndex|
+-------+------+-------+-------+-------+-------+------
+--------------------+----------+
|  Bream| 242.0|   23.2|   25.4|   30.0|  11.52|  4.02|
[23.2,25.4,30.0,1...|       1.0|
|  Bream| 290.0|   24.0|   26.3|   31.2|  12.48|4.3056|
[24.0,26.3,31.2,1...|       1.0|
|  Bream| 340.0|   23.9|   26.5|   31.1|12.3778|4.6961|
[23.9,26.5,31.1,1...|       1.0|
|  Bream| 363.0|   26.3|   29.0|   33.5|  12.73|4.4555|
[26.3,29.0,33.5,1...|       1.0|
|  Bream| 430.0|   26.5|   29.0|   34.0| 12.444| 5.134|
[26.5,29.0,34.0,1...|       1.0|
|  Bream| 450.0|   26.8|   29.7|   34.7|13.6024|4.9274|
[26.8,29.7,34.7,1...|       1.0|
|  Bream| 500.0|   26.8|   29.7|   34.5|14.1795|5.2785|
[26.8,29.7,34.5,1...|       1.0|
|  Bream| 390.0|   27.6|   30.0|   35.0|  12.67|  4.69|
[27.6,30.0,35.0,1...|       1.0|
|  Bream| 450.0|   27.6|   30.0|   35.1|14.0049|4.8438|
[27.6,30.0,35.1,1...|       1.0|
|  Bream| 500.0|   28.5|   30.7|   36.2|14.2266|4.9594|
[28.5,30.7,36.2,1...|       1.0|
|  Bream| 475.0|   28.4|   31.0|   36.2|14.2628|5.1042|
[28.4,31.0,36.2,1...|       1.0|
|  Bream| 500.0|   28.7|   31.0|   36.2|14.3714|4.8146|
[28.7,31.0,36.2,1...|       1.0|
|  Bream| 500.0|   29.1|   31.5|   36.4|13.7592| 4.368|
[29.1,31.5,36.4,1...|       1.0|
|  Bream| 340.0|   29.5|   32.0|   37.3|13.9129|5.0728|
[29.5,32.0,37.3,1...|       1.0|
|  Bream| 600.0|   29.4|   32.0|   37.2|14.9544|5.1708|
[29.4,32.0,37.2,1...|       1.0|
|  Bream| 600.0|   29.4|   32.0|   37.2| 15.438|  5.58|
[29.4,32.0,37.2,1...|       1.0|
|  Bream| 700.0|   30.4|   33.0|   38.3|14.8604|5.2854|
[30.4,33.0,38.3,1...|       1.0|
|  Bream| 700.0|   30.4|   33.0|   38.5| 14.938|5.1975|
[30.4,33.0,38.5,1...|       1.0|
|  Bream| 610.0|   30.9|   33.5|   38.6| 15.633|5.1338|
[30.9,33.5,38.6,1...|       1.0|
|  Bream| 650.0|   31.0|   33.5|   38.7|14.4738|5.7276|
[31.0,33.5,38.7,1...|       1.0|
+-------+------+-------+-------+-------+-------+------
+--------------------+----------+
only showing top 20 rows
```

```
pd.DataFrame(df.take(110), columns=df.columns).transpose()
```

```
train, test = df.randomSplit([0.7, 0.3], seed = 2018)
print("Training Dataset Count: " + str(train.count()))
print("Test Dataset Count: " + str(test.count()))
```

```
Training Dataset Count: 109
Test Dataset Count: 50
```

```
from pyspark.ml.classification import RandomForestClassifier

rf = RandomForestClassifier(featuresCol = 'features', labelCol =
'labelIndex')
rfModel = rf.fit(train)
predictions = rfModel.transform(test)
predictions.select('Length1','Length2','Length3','Height','Width',
'labelIndex', 'rawPrediction', 'prediction', 'probability').show(25)
```

```
+-------+-------+-------+-------+------+----------
+--------------------+----------+--------------------+
|Length1|Length2|Length3| Height| Width|labelIndex|
rawPrediction|prediction|         probability|
+-------+-------+-------+-------+------+----------
+--------------------+----------+--------------------+
|   23.2|   25.4|   30.0|  11.52|  4.02|       1.0|
[4.45457477883776...|       5.0|[0.22272873894188...|
|   27.6|   30.0|   35.1|14.0049|4.8438|       1.0|
[0.0,20.0,0.0,0.0...|       1.0|[0.0,1.0,0.0,0.0,...|
|   28.5|   30.7|   36.2|14.2266|4.9594|       1.0|
[0.0,20.0,0.0,0.0...|       1.0|[0.0,1.0,0.0,0.0,...|
|   28.7|   31.0|   36.2|14.3714|4.8146|       1.0|
[0.0,20.0,0.0,0.0...|       1.0|[0.0,1.0,0.0,0.0,...|
|   29.4|   32.0|   37.2| 15.438|  5.58|       1.0|
[0.0,20.0,0.0,0.0...|       1.0|[0.0,1.0,0.0,0.0,...|
|   31.0|   33.5|   38.7|14.4738|5.7276|       1.0|
[0.0,20.0,0.0,0.0...|       1.0|[0.0,1.0,0.0,0.0,...|
|   30.4|   33.0|   38.5| 14.938|5.1975|       1.0|
[0.0,20.0,0.0,0.0...|       1.0|[0.0,1.0,0.0,0.0,...|
|   36.2|   39.5|   45.3|18.7542|6.7497|       1.0|
[4.8,15.2,0.0,0.0...|       1.0|[0.24,0.76,0.0,0....|
|   35.0|   38.5|   44.0| 18.084| 6.292|       1.0|
[1.0,18.0,0.0,0.0...|       1.0|[0.05,0.9,0.0,0.0...|
|   37.4|   41.0|   45.9|18.6354|6.7473|       1.0|
[4.8,15.2,0.0,0.0...|       1.0|[0.24,0.76,0.0,0....|
|   13.5|   14.7|   16.5| 6.8475|2.3265|       5.0|
[7.43922949397113...|       5.0|[0.37196147469855...|
|   19.0|   20.7|   23.2| 8.5376|3.2944|       5.0|
[4.93596631965532...|       5.0|[0.24679831598276...|
|   19.0|   20.7|   23.2|  9.396|3.4104|       5.0|
[4.48401826770727...|       5.0|[0.22420091338536...|
|   21.2|   23.0|   25.8|10.3458|3.6636|       5.0|
[3.27689946058847...|       5.0|[0.16384497302942...|
|   15.0|   16.2|   17.2| 4.5924|2.6316|       0.0|
```

```
[13.3053009225425...|       0.0|[0.66526504612712...|
|   19.0|   21.0|   22.5| 5.6925|3.6675|       0.0|
[16.0332118109373...|       0.0|[0.80166059054686...|
|   20.5|   22.5|   24.0|  5.856| 3.624|       0.0|
[13.1094263371518...|       0.0|[0.65547131685759...|
|   20.5|   22.5|   24.0|  6.792| 3.624|       0.0|
[7.77384658253559...|       2.0|[0.38869232912677...|
|   21.0|   23.0|   24.5| 5.2185| 3.626|       0.0|
[14.7760930038185...|       0.0|[0.73880465019092...|
|   25.0|   26.5|   28.0|  7.168| 4.144|       0.0|
[12.4788532853391...|       0.0|[0.62394266426695...|
|   22.0|   24.0|   25.5|  7.293| 3.723|       0.0|
[11.1738465825355...|       0.0|[0.55869232912677...|
|   25.4|   27.5|   28.9| 7.2828|4.5662|       0.0|
[12.9187814730933...|       0.0|[0.64593907365466...|
|   32.0|   34.5|   36.5|10.2565|6.3875|       0.0|
[9.07126616162590...|       0.0|[0.45356330808129...|
|   34.6|   37.0|   39.3|10.5717|6.3666|       0.0|
[11.4284090187687...|       0.0|[0.57142045093843...|
|   34.0|   36.0|   38.3|10.6091|6.7408|       0.0|
[10.0712661616259...|       0.0|[0.50356330808129...|
+-------+-------+-------+-------+------+----------
+--------------------+----------+--------------------+
only showing top 25 rows
```

```
predictions.select("labelIndex", "prediction").show(10)
```

```
+----------+----------+
|labelIndex|prediction|
+----------+----------+
|       1.0|       5.0|
|       1.0|       1.0|
|       1.0|       1.0|
|       1.0|       1.0|
|       1.0|       1.0|
|       1.0|       1.0|
|       1.0|       1.0|
|       1.0|       1.0|
|       1.0|       1.0|
|       1.0|       1.0|
+----------+----------+
only showing top 10 rows
```

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

evaluator = MulticlassClassificationEvaluator(labelCol="labelIndex",
predictionCol="prediction")
accuracy = evaluator.evaluate(predictions)
```

```
print("Accuracy = %s" % (accuracy))
print("Test Error = %s" % (1.0 - accuracy))
```

```
Accuracy = 0.6973415520783942
Test Error = 0.3026584479216058
```

```
from pyspark.mllib.evaluation import MulticlassMetrics
from pyspark.sql.types import FloatType
import pyspark.sql.functions as F

preds_and_labels =
predictions.select(['prediction','labelIndex']).withColumn('labelIndex
', F.col('labelIndex').cast(FloatType())).orderBy('prediction')
preds_and_labels =
preds_and_labels.select(['prediction','labelIndex'])
metrics = MulticlassMetrics(preds_and_labels.rdd.map(tuple))
print(metrics.confusionMatrix().toArray())
```

```
/databricks/spark/python/pyspark/sql/context.py:165: FutureWarning:
Deprecated in 3.0.0. Use SparkSession.builder.getOrCreate() instead.
  warnings.warn(
[[13.  0.  1.  0.  0.  0.  0.]
 [ 0.  9.  0.  0.  0.  1.  0.]
 [ 8.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  7.  0.  0.  0.]
 [ 0.  0.  0.  0.  5.  0.  0.]
 [ 0.  0.  0.  0.  0.  4.  0.]
 [ 2.  0.  0.  0.  0.  0.  0.]]
```

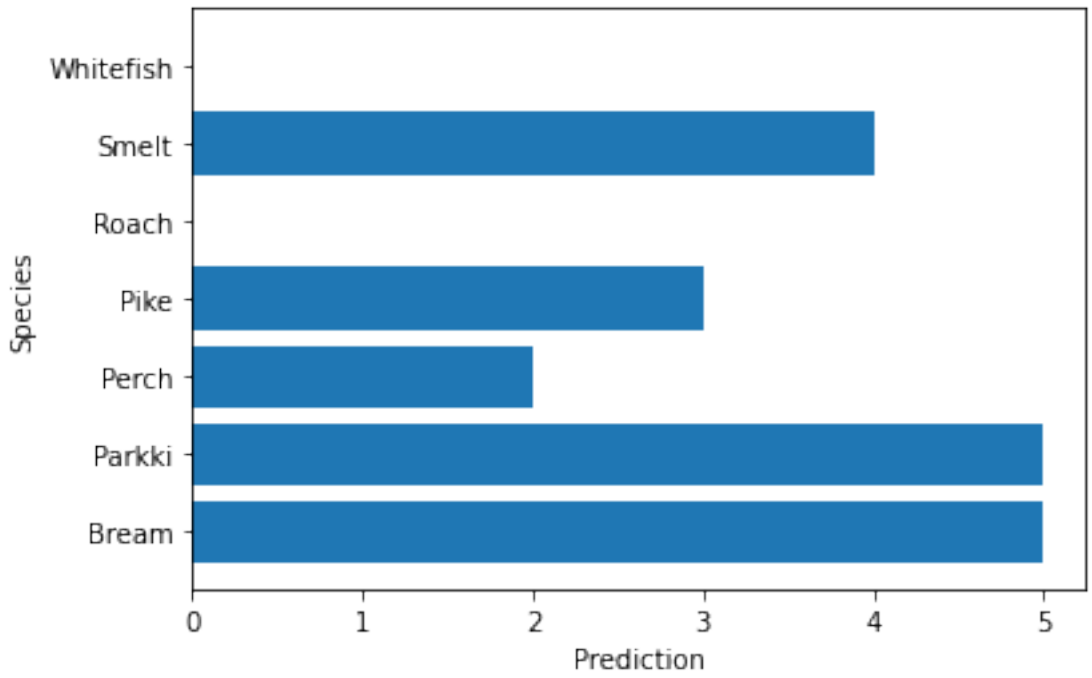
```
predictions.show()
```

```
+-------+------+-------+-------+-------+-------+------
+--------------------+----------+--------------------
+--------------------+----------+
|Species|Weight|Length1|Length2|Length3| Height| Width|
features|labelIndex|       rawPrediction|         probability|
prediction|
+-------+------+-------+-------+-------+-------+------
+--------------------+----------+--------------------
+--------------------+----------+
|  Bream| 242.0|   23.2|   25.4|   30.0|  11.52|  4.02|
[23.2,25.4,30.0,1...|       1.0|[4.45457477883776...|
[0.22272873894188...|       5.0|
|  Bream| 450.0|   27.6|   30.0|   35.1|14.0049|4.8438|
[27.6,30.0,35.1,1...|       1.0|[0.0,20.0,0.0,0.0...|
[0.0,1.0,0.0,0.0,...|       1.0|
|  Bream| 500.0|   28.5|   30.7|   36.2|14.2266|4.9594|
[28.5,30.7,36.2,1...|       1.0|[0.0,20.0,0.0,0.0...|
[0.0,1.0,0.0,0.0,...|       1.0|
|  Bream| 500.0|   28.7|   31.0|   36.2|14.3714|4.8146|
[28.7,31.0,36.2,1...|       1.0|[0.0,20.0,0.0,0.0...|
```

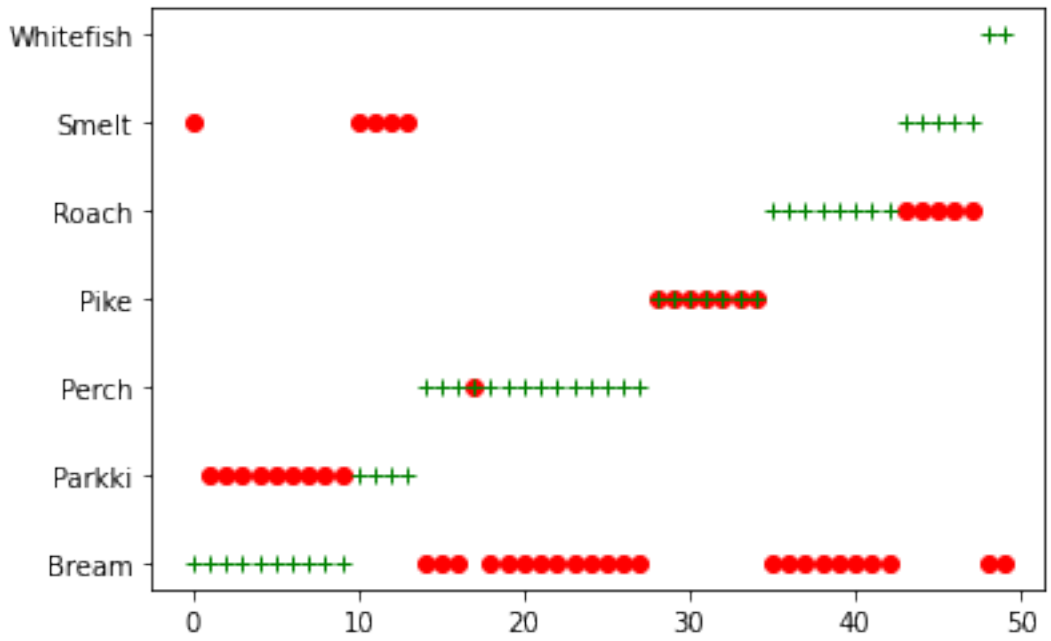
```
[0.0,1.0,0.0,0.0,...|       1.0|
|  Bream| 600.0|   29.4|   32.0|   37.2| 15.438|  5.58|
[29.4,32.0,37.2,1...|       1.0|[0.0,20.0,0.0,0.0...|
[0.0,1.0,0.0,0.0,...|       1.0|
|  Bream| 650.0|   31.0|   33.5|   38.7|14.4738|5.7276|
[31.0,33.5,38.7,1...|       1.0|[0.0,20.0,0.0,0.0...|
[0.0,1.0,0.0,0.0,...|       1.0|
|  Bream| 700.0|   30.4|   33.0|   38.5| 14.938|5.1975|
[30.4,33.0,38.5,1...|       1.0|[0.0,20.0,0.0,0.0...|
[0.0,1.0,0.0,0.0,...|       1.0|
|  Bream| 925.0|   36.2|   39.5|   45.3|18.7542|6.7497|
[36.2,39.5,45.3,1...|       1.0|[4.8,15.2,0.0,0.0...|
[0.24,0.76,0.0,0....|       1.0|
|  Bream| 955.0|   35.0|   38.5|   44.0| 18.084| 6.292|
[35.0,38.5,44.0,1...|       1.0|[1.0,18.0,0.0,0.0...|
[0.05,0.9,0.0,0.0...|       1.0|
|  Bream| 975.0|   37.4|   41.0|   45.9|18.6354|6.7473|
[37.4,41.0,45.9,1...|       1.0|[4.8,15.2,0.0,0.0...|
[0.24,0.76,0.0,0....|       1.0|
| Parkki|  55.0|   13.5|   14.7|   16.5| 6.8475|2.3265|
[13.5,14.7,16.5,6...|       5.0|[7.43922949397113...|
[0.37196147469855...|       5.0|
| Parkki| 140.0|   19.0|   20.7|   23.2| 8.5376|3.2944|
[19.0,20.7,23.2,8...|       5.0|[4.93596631965532...|
[0.24679831598276...|       5.0|
| Parkki| 170.0|   19.0|   20.7|   23.2|  9.396|3.4104|
[19.0,20.7,23.2,9...|       5.0|[4.48401826770727...|
[0.22420091338536...|       5.0|
| Parkki| 200.0|   21.2|   23.0|   25.8|10.3458|3.6636|
[21.2,23.0,25.8,1...|       5.0|[3.27689946058847...|
[0.16384497302942...|       5.0|
|  Perch|  51.5|   15.0|   16.2|   17.2| 4.5924|2.6316|
[15.0,16.2,17.2,4...|       0.0|[13.3053009225425...|
[0.66526504612712...|       0.0|
|  Perch| 125.0|   19.0|   21.0|   22.5| 5.6925|3.6675|
[19.0,21.0,22.5,5...|       0.0|[16.0332118109373...|
[0.80166059054686...|       0.0|
|  Perch| 130.0|   20.5|   22.5|   24.0|  5.856| 3.624|
[20.5,22.5,24.0,5...|       0.0|[13.1094263371518...|
[0.65547131685759...|       0.0|
|  Perch| 150.0|   20.5|   22.5|   24.0|  6.792| 3.624|
[20.5,22.5,24.0,6...|       0.0|[7.77384658253559...|
[0.38869232912677...|       2.0|
|  Perch| 150.0|   21.0|   23.0|   24.5| 5.2185| 3.626|
[21.0,23.0,24.5,5...|       0.0|[14.7760930038185...|
[0.73880465019092...|       0.0|
|  Perch| 218.0|   25.0|   26.5|   28.0|  7.168| 4.144|
[25.0,26.5,28.0,7...|       0.0|[12.4788532853391...|
[0.62394266426695...|       0.0|
+-------+------+-------+-------+-------+-------+------
```

```
+--------------------+----------+--------------------
+--------------------+----------+
only showing top 20 rows
```

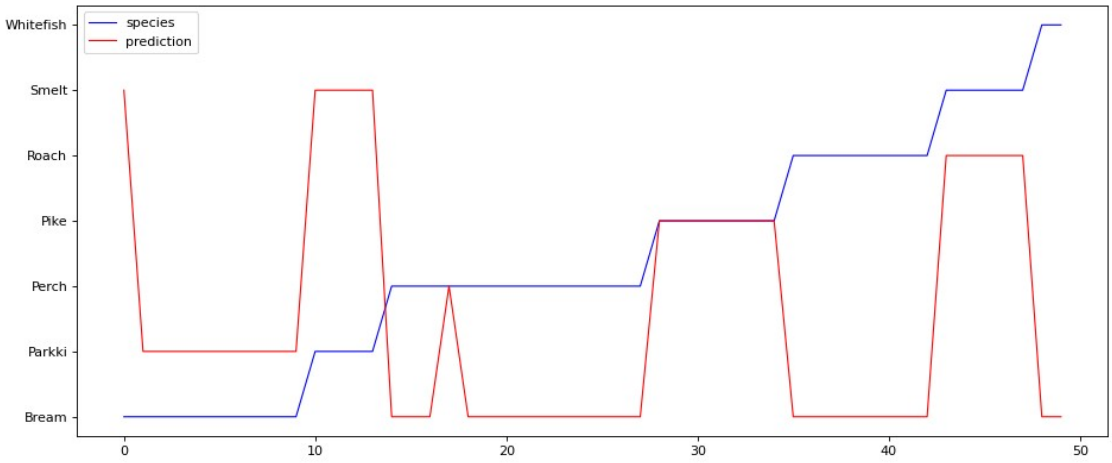
```
predictions.createOrReplaceTempView("df_sql")

fish = spark.sql("Select Species from df_sql")
fish = fish.rdd.map(lambda row : row.Species).collect()

pre = spark.sql("Select prediction from df_sql")
pre = pre.rdd.map(lambda row : row.prediction).collect()

plt.barh(fish,pre)

plt.xlabel("Prediction")
plt.ylabel("Species")


plt.show()
```




```
plt.plot(pre,'ro',fish,'g+')

Out[209]: [<matplotlib.lines.Line2D at 0x7f21f0f28850>,
 <matplotlib.lines.Line2D at 0x7f21f0f28af0>]
```

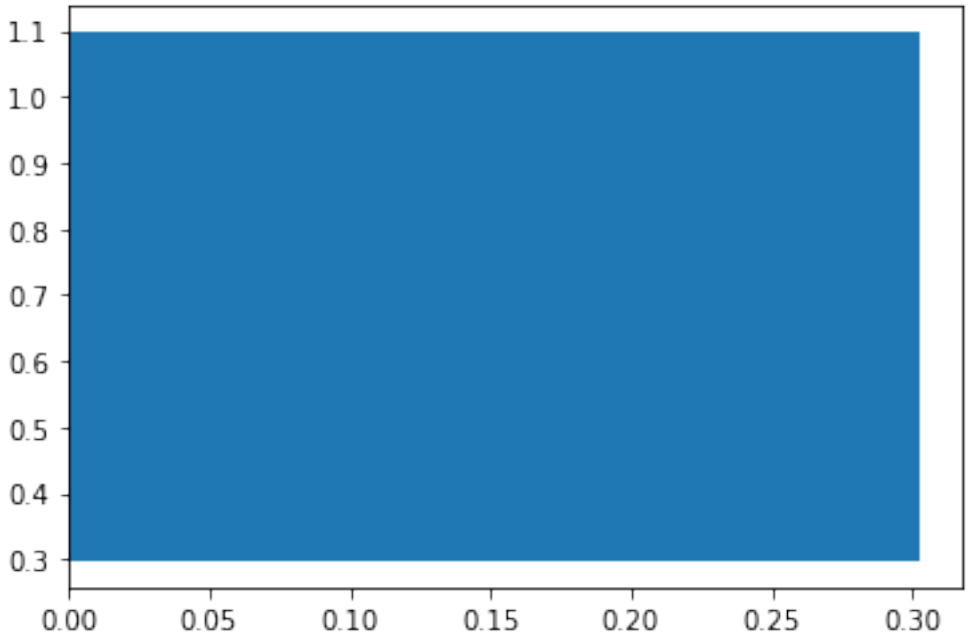



```
  fig,ax = plt.subplots(num=None,figsize=(14,6), dpi=80,
facecolor='w', edgecolor='k')
size=len(pre)
ax.plot(range(0,size), fish, color='blue', animated=True, linewidth=1,
label='species')
ax.plot(range(0,size), pre, color='red', animated=True, linewidth=1,
label='prediction')
legend=ax.legend()
```




```
test_error=1-accuracy
plt.barh(accuracy,test_error)
```

```
Out[211]: <BarContainer object of 1 artists>
```


1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.00
0.05
0.10
0.15
0.20
0.25
0.30