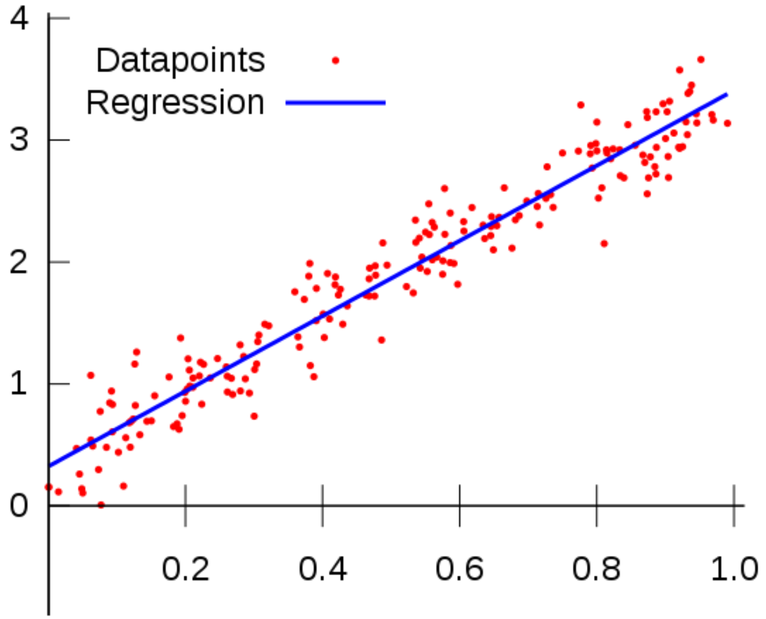
Practice 2

Regression

Problem

- Construct spark environment in your local computer and use three regression methods: Linear least squares, Lasso and Ridge regression
- Note that Ridge and Lasso regression have regularization term, so they may be able to avoid overfitting problem. But **Least Square regression can't.**



- Use predefined function in pyspark.**mllib.regression**

Dataset

- **Artificial dataset from pyspark tutorial**
  - This data is given from the reference link on the bottom
  - You can see whatever you want about pyspark mllib in this link.

- **Dataset format**
  - The first number is target
  - The remains are features

- **You can download the training and test dataset on i-campus**

https://github.com/apache/spark/blob/master/data/mllib/ridge-data/lpsa.data

Practice 2

1. **Use predefined classes in *pyspark.mllib.regression : LinearRegressionWIthSGD(), RidgeRegressionWithSGD(), LassoWithSGD().*** ***Please refer to hyperlinks below***

   Parameters for each method

   - LinearRegressionWIthSGD : iteration = 100, step = 0. 1
   - RidgeRegressionWithSGD : iteration = 100, step = 0.001, regParam = 0.01
   - LassoWithSGD : iteration = 100, step = 0.001, regParam = 0.01

2. **After training the models, calculate the root mean square error(RMSE) using all data points for each algorithm.**
3. **Write a simple report with RMSE of each algorithm.**

https://spark.apache.org/docs/latest/mllib-linear-methods.html#linear-least-squares-lasso-and-ridge-regression

https://spark.apache.org/docs/latest/api/python/pyspark.mllib.html#pyspark.mllib.regression

Submission

1. Submit “**result.txt**” file which includes Root Mean Squared Error(RMSE) of **Least Square**, **Ridge** and **Lasso regression**.
2. You must write the result of applying your trained model to **training data points** and **test data points**.
3. Your **results.txt** file must be like following.

```
RMSE train / test
LEAST 2.0891, 4.4972
RIDGE 2.2646, 4.0287
LASSO 2.2646, 4.0287
```

<Windows>

```
RMSE train / test
LEAST 2.0891, 4.4972
RIDGE 2.2646, 4.0287
LASSO 2.2646, 4.0287
```

<Linux>

Solution

- Import package

```
from pyspark import SparkConf, SparkContext
from pyspark.mllib.regression import LabeledPoint, LinearRegressionWithSGD
from pyspark.mllib.regression import RidgeRegressionWithSGD, LassoWithSGD
import math
```

Import the Spark Package

- Initialize a SparkContext

```
conf = SparkConf()
conf.set("spark.master", "local")
sc = SparkContext(conf=conf)
```

Configure Spark with SparkConf
- spark's master as local

- Parse data to LabeledPoint

```
def parsePoint(line):
    try:
        values = [float(x) for x in line.replace(',',' ').split(' ')]
        return LabeledPoint(values[0], values[1:])
    except:
        return None
```

Make data to have label attribute, and feature attribute

Solution

- Create RDDs (Import data) & Parse the data

```
data = sc.textFile("train.data")
trainData = data.map(parsePoint)

data = sc.textFile("test.data")
testData = data.map(parsePoint)
```

Create RDDs
textFile(): load data from an external storage

Transform RDDs

- Least sqaure regression

```
# Least Square Regression
model_least = LinearRegressionWithSGD.train(trainData, iterations=100, step=0.1)

valuesAndPreds = testData.map(lambda p:
                              (p.label, model_least.predict(p.features)))
MSE = valuesAndPreds.map(lambda vp:
                         (vp[0]-vp[1])**2).reduce(lambda x, y: x + y) / valuesAndPreds.count()
test_cnt = valuesAndPreds.count()
least_RMSE_test = math.sqrt(MSE)

valuesAndPreds = trainData.map(lambda p:
                               (p.label, model_least.predict(p.features)))
MSE = valuesAndPreds.map(lambda vp:
                         (vp[0]-vp[1])**2).reduce(lambda x, y: x + y) / valuesAndPreds.count()
train_cnt = valuesAndPreds.count()
least_RMSE_train = math.sqrt(MSE)
```

Train the model using train RDDs that have label.

Predict test RDDs label

Solution

- Ridge regression(same with linear regression)

Train the model using train RDDs that have label.

Predict test RDDs label

Predict train RDDs label

```
# Ridge Regression
model_ridge = RidgeRegressionWithSGD.train(trainData, iterations=100, step=0.001, regParam=0.01)

valuesAndPreds = testData.map(lambda p:
                              (p.label, model_ridge.predict(p.features)))
MSE = valuesAndPreds.map(lambda vp:
                         (vp[0]-vp[1])**2).reduce(lambda x, y: x + y) / valuesAndPreds.count()
test_cnt = valuesAndPreds.count()
ridge_RMSE_test = math.sqrt(MSE)

valuesAndPreds = trainData.map(lambda p:
                               (p.label, model_ridge.predict(p.features)))
MSE = valuesAndPreds.map(lambda vp:
                         (vp[0]-vp[1])**2).reduce(lambda x, y: x + y) / valuesAndPreds.count()
train_cnt = valuesAndPreds.count()
ridge_RMSE_train = math.sqrt(MSE)
```

Solution

- **Lasso regression(same with linear regression)**

```
# Lasso Regression
model_lasso = LassoWithSGD.train(trainData, iterations=100, step=0.001, regParam=0.01)

valuesAndPreds = testData.map(lambda p:
                              (p.label, model_lasso.predict(p.features)))
MSE = valuesAndPreds.map(lambda vp:
                         (vp[0]-vp[1])**2).reduce(lambda x, y: x + y) / valuesAndPreds.count()
test_cnt = valuesAndPreds.count()
lasso_RMSE_test = math.sqrt(MSE)

valuesAndPreds = trainData.map(lambda p:
                               (p.label, model_lasso.predict(p.features)))
MSE = valuesAndPreds.map(lambda vp:
                         (vp[0]-vp[1])**2).reduce(lambda x, y: x + y) / valuesAndPreds.count()
train_cnt = valuesAndPreds.count()
lasso_RMSE_train = math.sqrt(MSE)
```

Solution

➢ Save the result

```
f = open("result.txt","w")
f.write('RMSE train / test\n')
f.write('LEAST {:.4f}, {:.4f}\n'.format(least_RMSE_train, least_RMSE_test))
f.write('RIDGE {:.4f}, {:.4f}\n'.format(ridge_RMSE_train, ridge_RMSE_test))
f.write('LASSO {:.4f}, {:.4f}\n'.format(lasso_RMSE_train, lasso_RMSE_test))
```

```
sc.stop()
```

Must stop SparkContext

➢ Result

- As you can see, the results are same both on Windows and Linux

RMSE train / test
LEAST 2.0891, 4.4972
RIDGE 2.2646, 4.0287
LASSO 2.2646, 4.0287

<Windows>

```
RMSE train / test
LEAST 2.0891, 4.4972
RIDGE 2.2646, 4.0287
LASSO 2.2646, 4.0287
```

<Linux>