


# M

## ml-intro-spark-regression

In this project we do a brief introduction to ML by using doing linear regression with python and Apache spark. We also use Spark SQL to do descriptive analysis of the data.

 (/MachineLearning/ml-intro-spark-regression/edit/master/README.md)

## Introduction to Machine using Spark

Project Contact: Mohinder Dick (mailto:dickm@upmc.edu)

The goal of this repository is to house Jupyter notebooks that demonstrate simple machine learning tasks using Spark.

- Java, preferably version 1.7 or later
- Python, preferably the Anaconda stack. See installation instructions here (<http://docs.continuum.io/anaconda/install>).
- Spark, version 1.4.1 or later. See installation instructions downloading section here (<http://spark.apache.org/docs/latest/>).
- Hadoop, version 2.6.0 or later. Get tar and untar from here (<http://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-2.6.0/hadoop-2.6.0.tar.gz>).

## ML Palooza

For the ML palooza you can run the Apache Spark against our YARN cluster. We will support R access soon. Follow the instructions below to set up Spark and Hadoop for access. For other stacks you follow the instructions below to set up Hadoop. At the very least you can copy the files locally Ensure you have met the prerequisites.

Your end goal for Spark is to open an interactive session in a browser-based IDE for PySpark called Jupyter.

### Overview

You are trying to predict the length-of-stay (LOS) given the data in the file visit\_train\_panda.csv.

\$100 Amazon Gift Card

March 4, 2016

### Data

The HDFS path is: /palooza/data/visit\_train\_panda.csv.

The data has the followings. I describe the fields I know:

- VisitID - Identifier for patient visit.
- Hospital - Admitting hospital.
- Dept\_Code - department code.
- PaymentType - I am guessing a payment type for visit.
- Age - Age of the patient in years.
- Race - De-identified race of the patient.
- Gender - Gender ("M" - male, "F" - female)
- FC - ?
- ArriveDate - Date of admission.
- DischargeDate - Date of discharge
- LOS - length of patient stay in days.
- DXCODE - Diagnosis code.
- Description - Description of diagnosis
- DispenseID - ?
- DOC - ?

## Submitting Solutions

Submit your solutions in a well-named branch off of this repo. Update the README.md with any additional set up instructions for running your solution on an arbitrary data set with same schema.

## Palooza Environment Set up

### Other Stacks (Non-Spark)

If you are not connecting to the Spark cluster, you can use any API that allows you to access a HTTP endpoint. We prefer that you *stream*, not download, the file. The data is very sensitive.

You can do so by accessing this REST end-point:

```
http://sparkd104:50070/webhdfs/v1/palooza/data/visit_train_panda.csv?op=OPEN
```

```
dataFrame = read.csv("http://sparkd104:50070/webhdfs/v1/palooza/data/visit_train_panda.csv?op=OPEN")
```

### Spark Setup

#### Initial Setup

Set up Environment variables

- Add JAVA\_HOME
- Add SPARK\_HOME environment variable and set to the root directory at the path you copied Apache Spark (e.g. /opt/spark-1.4.1-bin-hadoop2.6).
- Add HADOOP\_HOME environment variable and set to the root directory at the path you copied Hadoop (e.g. /opt/hadoop-2.6.2)
- Add HADOOP\_CONF\_DIR environment variable and set to \$HADOOP\_HOME/conf
- Add the path to the anaconda python folder to Path variable.
- Add the entries in hosts from git to your local hosts

#### Hadoop setup

Copy conf directory from git to \$HADOOP\_HOME (i.e this will resolve \$HADOOP\_CONF\_DIR).

#### Verify Access to Spark

Hadoop - run `hadoop fs -ls /`. You should get a list of directories including palooza. Spark:

- In bash shell type `$SPARK_HOME/bin/pyspark`
- In the Python interactive prompt run. You should get the result 100: `sc.parallelize(range(100)).count()`
- Spark System test: Run the testYARN.py by running the following in a bash prompt (it should complete with out error):  
`$SPARK_HOME/bin/spark-submit --master yarn-client testYARN.py`

#### Run Spark in Jupyter Notebook

You will most likely want to run spark using a notebook. You can start with the notebook IntroToMLwithPySpark.ipynb.

- Run the following command then connect via a browser to work interactively with Python Spark. You can tweak the memory values.:  
`PATH=$PATH:<anaconda python path> PYSARK_PYTHON=<anaconda python path>/bin/python SPARK_EXECUTOR_MEMORY="1G"`

#### Initial Setup

Set up Environment variables

- Add JAVA\_HOME
- Add SPARK\_HOME environment variable and set to the root directory at the path you copied Apache Spark (e.g. C:\spark-1.4.1-bin-hadoop2.6).
- Add HADOOP\_HOME environment variable and set to the root directory at the path you copied Hadoop (e.g. C:\hadoop-2.6.2)
- Add HADOOP\_CONF\_DIR environment variable and set to \$HADOOP\_HOME\etc\hadoop
- Add the path to the anaconda python folder to your PATH variable. It may already be there if you chose option in the install of anaconda.
- Add the entries in hosts from git to your local hosts file (i.e. C:\Windows\System32\drivers\etc\hosts)

#### Hadoop setup

Copy contents of the conf folder from git (about 30 files) to path defined in \$HADOOP\_CONF\_DIR in the initial setup.

### Verify Access to Spark

Hadoop - run `hadoop fs -ls /`. You should get a list of directories including `palooza`. Spark:

- In command prompt run `%SPARK_HOME%\bin\pyspark`
- In the Python interactive prompt run. You should get the result 100: `sc.parallelize(range(100)).count()`
- Spark System test: Run the `testYARN.py` by running the following in a bash prompt (it should complete with out error):  
`%SPARK_HOME%\bin\spark-submit --master yarn-client testYARN.py`

### Run Spark in Jupyter Notebook

You will most likely want to run spark using a Jupyter notebook. You can start with the notebook `IntroToMLwithPySpark.ipynb`.

- Set the following environment variables for the session or user: `*PYSPARK_PYTHON=/opt/anaconda/bin/python`
- `PYSPARK_DRIVER_PYTHON="ipython"`
- `PYSPARK_DRIVER_PYTHON_OPTS="notebook"`
- Run the following command. You default browser should launch. If not connect to localhost using port indicated. Port 8888 is the default. You can tweak the memory values.:  
`%SPARK_HOME%\bin\pyspark --master "yarn-client" --conf spark.executor.memory=1g --conf spark.driver.memory=1g`

You have Owner access to this project.