PROJECT ON HADOOP & SPARK

CUSTOMER CHURN ANALYSIS

CSIT DEPARTMENT

by

Priyanshu Verma.

CONTENT

=>INTRODUCTION

=>TECHNOLOGY WE USED

- WHAT IS HADOOP
- CREATE A VIRTUAL MACHINE WITH A CENTOS OPERATING SYSTEM
- INSTALL JDK AND HADOOP ON THE VIRTUAL MACHINE
- CONFIGURE AND EDIT BASHRC FILES OF HADOOP
- RUN ALL THE DEMONS
- INSTALL SPARK IN A VIRTUAL MACHINE
- INSATLL PYSPARK

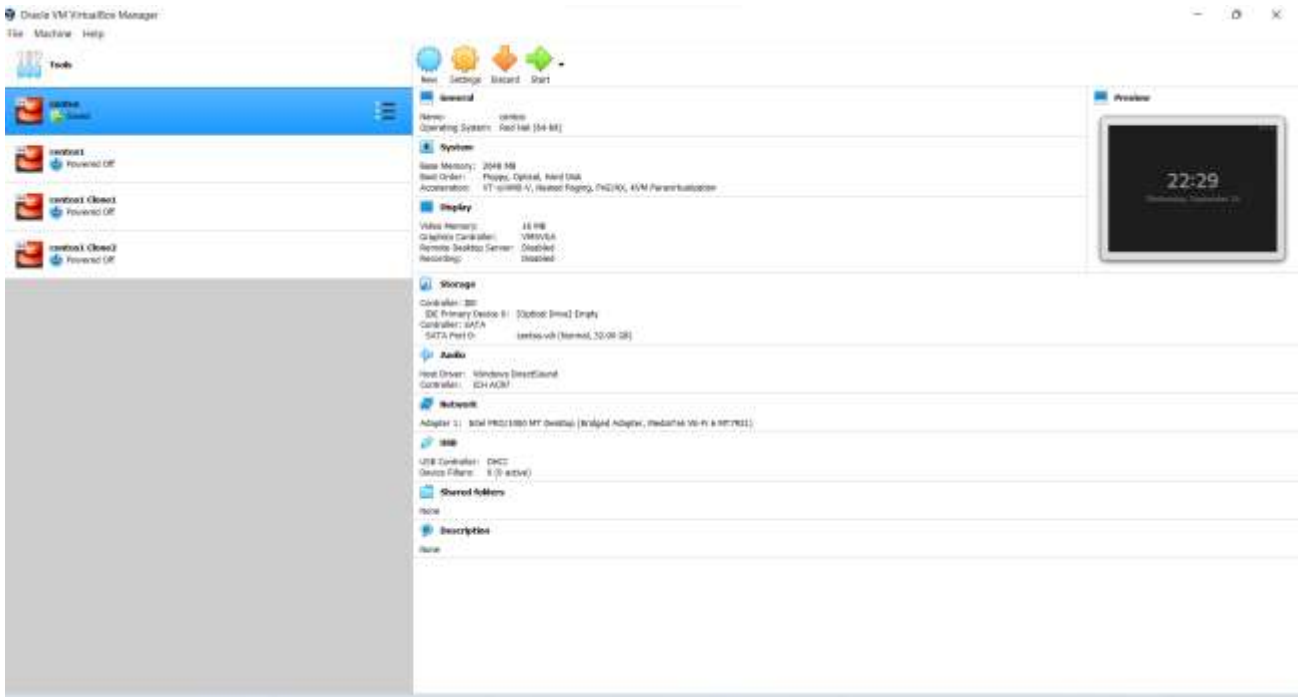
- INSTALL THE PIP FILE
- PIP INSTALL JUPYTER NOTEBOOK.
- NOW WE USED MACHINE LEARNING WITH PYSPARK FOR EXTRACTION OF THE DATASET.

Introduction

Customer Churn analysis measures **the rate at which customers quit the product, site, or service**. It answers the questions “Are we losing customers?” and “If so, how?” to allow teams to take action. Lower churn rates lead to happier customers, larger margins, and higher profits. The Customer Churn analysis is done by using Hadoop with the Centos operating system and also we used Pyspark.

What is Hadoop

Hadoop is an open-source framework that allows to store and process big data in a distributed environment across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.



Create virtual machine with centos operating system

```
javapointers@localhost:/usr/local/hadoop/hadoop-2.10.0
File  Edit  View  Search  Terminal  Help
  GNU nano 2.9.8                    /home/javapointers/.bashrc

# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific environment
export JAVA_HOME=/usr/java/jdk1.8.0_131
export HADOOP_HOME=/usr/local/hadoop/hadoop-2.10.0
export PATH=$PATH:$HADOOP_HOME/bin

PATH="$HOME/.local/bin:$HOME/bin:$PATH"
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
```

Install jdk and Hadoop in virtual machine

CONFIGURE AND EDIT BASHRC FILES OF HADOOP

```
# .bashrc

: User specific aliases and functions

lias rm='rm -i'
lias cp='cp -i'
ias mv='mv -i'

Source global definitions
 [ -f /etc/bashrc ]; then
      . /etc/bashrc


port HADOOP_HOME=/home/hadoop/hadoop
port HADOOP_INSTALL=$HADOOP_HOME
port HADOOP_MAPRED_HOME=$HADOOP_HOME
port HADOOP_COMMON_HOME=$HADOOP_HOME
port HADOOP_HDFS_HOME=$HADOOP_HOME
port YARN_HOME=$HADOOP_HOME
port HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/na
port PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin

-- INSERT --
```

RUN ALL THE DEMONS

```
edureka@localhost:~/hadoop-2.7.3/sbin
File Edit View Search Terminal Help
[edureka@localhost sbin]$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/edureka/hadoop-2.7.3/logs/mapred-edureka-h
istoryserver-localhost.localdomain.out
[edureka@localhost sbin]$ jps
22113 NameNode
22310 ResourceManager
22694 JobHistoryServer
22727 Jps
22206 DataNode
22559 NodeManager
[edureka@localhost sbin]$
```

WHAT IS SPARK?

Spark is an open source framework focused on **interactive query, machine learning, and real-time workloads**. It does not have its own storage system, but runs analytics on other storage systems like HDFS, or other popular stores like Amazon Redshift, Amazon S3, Couchbase, Cassandra, and others.

What is a PySpark?

PySpark is **the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing**. If you're already familiar with Python and libraries such as Pandas, then PySpark is a good language to learn to create more scalable analyses and pipelines.

Installation of Spark

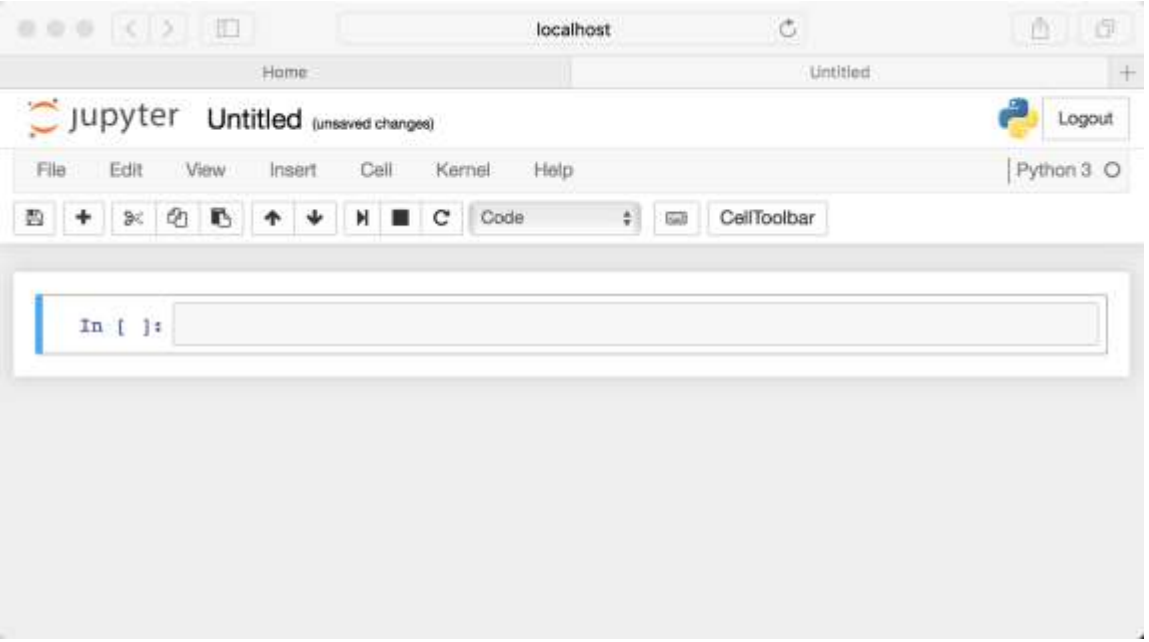
codegyani@ubuntu64server: /spark-install

```
codegyani@ubuntu64server:/spark-install$ spark-shell
19/04/12 10:07:23 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve
l(newLevel).
Spark context Web UI available at http://ubuntu64server:4040
Spark context available as 'sc' (master = local[*], app id = local-1555078159039
).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.1
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_60)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

JUPYTER NOTEBOOK



Then we apply some basic machine learning algorithms to find the output we need.

- Vector Assembler
- Logistic Regression
- Binary classification Evaluator
- Multi-classification Evaluator
- Confusion matrix / Classification matrix

- SOME OUTPUT ARE……..

```
[12]  df_final.show()

...
+--------------------+-----+
|            features|churn|
+--------------------+-----+
|[42.0,11066.8,0.0...|    1|
|[41.0,11916.22,0....|    1|
|[38.0,12884.75,0....|    1|
|[42.0,8010.76,0.0...|    1|
|[37.0,9191.58,0.0...|    1|
|[48.0,10356.02,0....|    1|
|[44.0,11331.58,1....|    1|
|[32.0,9885.12,1.0...|    1|
|[43.0,14062.6,1.0...|    1|
|[40.0,8066.94,1.0...|    1|
|[30.0,11575.37,1....|    1|
|[45.0,8771.02,1.0...|    1|
|[45.0,8988.67,1.0...|    1|
|[40.0,8283.32,1.0...|    1|
|[41.0,6569.87,1.0...|    1|
|[38.0,10494.82,1....|    1|
|[45.0,8213.41,1.0...|    1|
|[43.0,11226.88,0....|    1|
|[53.0,5515.09,0.0...|    1|
|[46.0,8046.4,1.0,...|    1|
+--------------------+-----+
only showing top 20 rows
```

Figure 7: Feature/churn selection of Vector assembler

SOME OUTPUTS ARE....

```
+--------------------+-----+--------------------+--------------------+----------+
|            features|churn|       rawPrediction|         probability|prediction|
+--------------------+-----+--------------------+--------------------+----------+
|[22.0,11254.38,1....|  0.0|[4.96050524236784...|[0.99303940400139...|       0.0|
|[27.0,8628.8,1.0,...|  0.0|[5.90597469051668...|[0.99728426899703...|       0.0|
|[28.0,8670.98,0.0...|  0.0|[8.19872388612018...|[0.99972507132363...|       0.0|
|[28.0,9090.43,1.0...|  0.0|[1.70655875854888...|[0.84638940648796...|       0.0|
|[28.0,11128.95,1....|  0.0|[4.43696894638154...|[0.98830660639038...|       0.0|
|[28.0,11204.23,0....|  0.0|[1.81406789469421...|[0.85985279779525...|       0.0|
|[28.0,11245.38,0....|  0.0|[3.69189769600253...|[0.97568147324479...|       0.0|
|[29.0,5900.78,1.0...|  0.0|[4.59258813963566...|[0.98997490494809...|       0.0|
|[29.0,8688.17,1.0...|  1.0|[2.96650688314298...|[0.95103787662661...|       0.0|
|[29.0,9378.24,0.0...|  0.0|[4.92859535749842...|[0.99281533194024...|       0.0|
|[29.0,10203.18,1....|  0.0|[4.01265512959848...|[0.98223595571025...|       0.0|
|[29.0,11274.46,1....|  0.0|[4.78549371666795...|[0.99171914494550...|       0.0|
|[29.0,13240.01,1....|  0.0|[7.03746026219900...|[0.99912241630757...|       0.0|
|[29.0,13255.05,1....|  0.0|[4.30395241860729...|[0.98666518428846...|       0.0|
|[30.0,6744.87,0.0...|  0.0|[3.71004655959537...|[0.97610839642025...|       0.0|
|[30.0,7960.64,1.0...|  1.0|[3.57503617850392...|[0.97274900827954...|       0.0|
|[30.0,8403.78,1.0...|  0.0|[6.44804766880066...|[0.99841889297270...|       0.0|
|[30.0,8677.28,1.0...|  0.0|[4.41655478245109...|[0.98806831982257...|       0.0|
|[30.0,8874.83,0.0...|  0.0|[3.23928343658757...|[0.96228611290728...|       0.0|
|[30.0,10183.98,1....|  0.0|[3.10250401931621...|[0.95699591531282...|       0.0|
+--------------------+-----+--------------------+--------------------+----------+
only showing top 20 rows
```

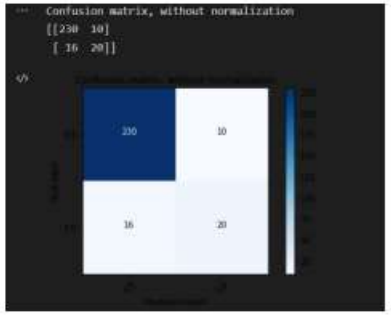
Figure 8: Prediction and Raw prediction of Logistic regression

```
[24]  auc

      0.7569444444444445
```

Figure 10: Accuracy of BinaryClassificationEvaluater

SOME OUTPUTS ARE.....



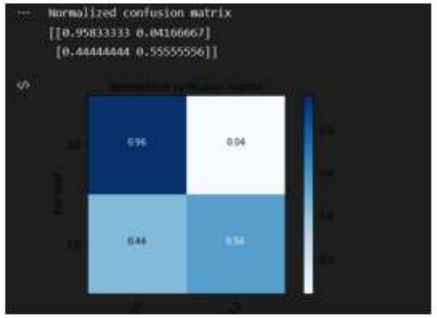


Figure 15: Confusion matrix without Normalisation

Figure 16: Confusion matrix with Normalisation

```
array([[230,  10],
       [ 16,  20]])
```

Figure 17: Array form of data in confusion matrix

THANK YOU