PLEASE NOTE: Please run this notebook OUTSIDE a Spark notebook as it should run in a plain Default Python 3.6 Free Environment

This is the last assignment for the Coursera course "Advanced Machine Learning and Signal Processing"

Just execute all cells one after the other and you are done - just note that in the last one you should update your email address (the one you've used for coursera) and obtain a submission token, you get this from the programming assignment directly on coursera.

Please fill in the sections labelled with "###YOUR_CODE_GOES_HERE###"

The purpose of this assignment is to learn how feature engineering boosts model performance. You will apply Discrete Fourier Transformation on the accelerometer sensor time series and therefore transforming the dataset from the time to the frequency domain.

After that, you'll use a classification algorithm of your choice to create a model and submit the new predictions to the grader. Done.

In [1]:
```
from IPython.display import Markdown, display
def printmd(string):
    display(Markdown('# <span style="color:red">'+string+'</span>'))


if ('sc' in locals() or 'sc' in globals()):
    printmd('<<<<<!!!!! It seems that you are running in a IBM Watson Studio Apache Spark Notebook. Please run it in an IBM Watson Studio Default Runtime (without Apache Spark) !!
```

In [2]:
```
!pip install pyspark==2.4.5
```

```
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes ins
tead
  from cryptography.utils import int_from_bytes
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/util.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Collecting pyspark==2.4.5
  Downloading pyspark-2.4.5.tar.gz (217.8 MB)
     |████████████████████████████████| 217.8 MB 9.6 kB/s  eta 0:00:01
Collecting py4j==0.10.7
  Downloading py4j-0.10.7-py2.py3-none-any.whl (197 kB)
     |████████████████████████████████| 197 kB 50.0 MB/s eta 0:00:01
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-2.4.5-py2.py3-none-any.whl size=218257927 sha256=6ac15d0e92af2c9a0408ba428f574ad9a4be039cdc58b0bf026d9d3ab7e5df95
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/01/c0/03/1c241c9c482b647d4d99412a98a5c7f87472728ad41ae55e1e
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.7 pyspark-2.4.5
```

In [3]:
```
!pip install https://github.com/IBM/coursera/blob/master/systemml-1.3.0-SNAPSHOT-python.tar.gz?raw=true
```

```
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes ins
tead
  from cryptography.utils import int_from_bytes
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/util.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Collecting https://github.com/IBM/coursera/blob/master/systemml-1.3.0-SNAPSHOT-python.tar.gz?raw=true
  Downloading https://github.com/IBM/coursera/blob/master/systemml-1.3.0-SNAPSHOT-python.tar.gz?raw=true (9.9 MB)
     |████████████████████████████████| 9.9 MB 22.7 MB/s eta 0:00:01
Requirement already satisfied: numpy>=1.8.2 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from systemml==1.3.0) (1.18.5)
Requirement already satisfied: scipy>=0.15.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from systemml==1.3.0) (1.5.0)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from systemml==1.3.0) (1.0.5)
Requirement already satisfied: scikit-learn in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from systemml==1.3.0) (0.23.1)
Requirement already satisfied: Pillow>=2.0.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from systemml==1.3.0) (7.2.0)
Requirement already satisfied: python-dateutil>=2.6.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pandas->systemml==1.3.0) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pandas->systemml==1.3.0) (2020.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from scikit-learn->systemml==1.3.0) (2.1.0)
Requirement already satisfied: joblib>=0.11 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from scikit-learn->systemml==1.3.0) (0.16.0)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from python-dateutil>=2.6.1->pandas->systemml==1.3.0) (1.15.0)
Building wheels for collected packages: systemml
  Building wheel for systemml (setup.py) ... done
  Created wheel for systemml: filename=systemml-1.3.0-py3-none-any.whl size=9882972 sha256=c82d9948f84f100ff2f3cc27c7a3526a121372214c3a6b08e881dfb0ca162dc8
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/ed/96/15/1042ed0087d53c21a17788d99d5581169482cfe683f1f6e60a
Successfully built systemml
Installing collected packages: systemml
Successfully installed systemml-1.3.0
```

In [4]:
```
from pyspark import SparkContext, SparkConf
from pyspark.sql import SQLContext, SparkSession
from pyspark.sql.types import StructType, StructField, DoubleType, IntegerType, StringType
sc = SparkContext.getOrCreate(SparkConf().setMaster("local[*]"))
from pyspark.sql import SparkSession
spark = SparkSession \
    .builder \
    .getOrCreate()
```

So the first thing we need to ensure is that we are on the latest version of SystemML, which is 1.3.0 (as of 20th March'19) Please use the code block below to check if you are already on 1.3.0 or higher. 1.3 contains a necessary fix, that's we are running against the SNAPSHOT

In [5]:
```
!mkdir -p /home/dsxuser/work/systemml
```

```
mkdir: cannot create directory ‘/home/dsxuser’: Permission denied
```

In [6]:
```
from systemml import MLContext, dml
ml = MLContext(spark)
ml.setConfigProperty("sysml.localtmpdir", "mkdir /home/dsxuser/work/systemml")
print(ml.version())

if not ml.version() == '1.3.0-SNAPSHOT':
    raise ValueError('please upgrade to SystemML 1.3.0, or restart your Kernel (Kernel->Restart & Clear Output)')
```

```
1.3.0-SNAPSHOT
```

In [7]:
```
!wget https://github.com/IBM/coursera/blob/master/coursera_ml/shake.parquet?raw=true
!mv shake.parquet?raw=true shake.parquet
```

```
--2021-04-16 20:00:27--  https://github.com/IBM/coursera/blob/master/coursera_ml/shake.parquet?raw=true
```

```
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://github.com/IBM/skillsnetwork/blob/master/coursera_ml/shake.parquet?raw=true [following]
--2021-04-16 20:00:27--  https://github.com/IBM/skillsnetwork/blob/master/coursera_ml/shake.parquet?raw=true
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/IBM/skillsnetwork/raw/master/coursera_ml/shake.parquet [following]
--2021-04-16 20:00:27--  https://github.com/IBM/skillsnetwork/raw/master/coursera_ml/shake.parquet
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/IBM/skillsnetwork/master/coursera_ml/shake.parquet [following]
--2021-04-16 20:00:27--  https://raw.githubusercontent.com/IBM/skillsnetwork/master/coursera_ml/shake.parquet
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.108.133, 185.199.109.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 74727 (73K) [application/octet-stream]
Saving to: ‘shake.parquet?raw=true’

shake.parquet?raw=t 100%[===================>]  72.98K  --.-KB/s    in 0.002s

2021-04-16 20:00:28 (47.0 MB/s) - ‘shake.parquet?raw=true’ saved [74727/74727]
```

Now it's time to read the sensor data and create a temporary query table.

In [8]:
```
df=spark.read.parquet('shake.parquet')
```

In [9]:
```
df.show()
```

```
+-----+---------+-----+-----+-----+
|CLASS| SENSORID|    X|    Y|    Z|
+-----+---------+-----+-----+-----+
|    2| qqqqqqqq| 0.12| 0.12| 0.12|
|    2|aUniqueID| 0.03| 0.03| 0.03|
|    2| qqqqqqqq|-3.84|-3.84|-3.84|
|    2| 12345678| -0.1| -0.1| -0.1|
|    2| 12345678|-0.15|-0.15|-0.15|
|    2| 12345678| 0.47| 0.47| 0.47|
|    2| 12345678|-0.06|-0.06|-0.06|
|    2| 12345678|-0.09|-0.09|-0.09|
|    2| 12345678| 0.21| 0.21| 0.21|
|    2| 12345678|-0.08|-0.08|-0.08|
|    2| 12345678| 0.44| 0.44| 0.44|
|    2|    gholi| 0.76| 0.76| 0.76|
|    2|    gholi| 1.62| 1.62| 1.62|
|    2|    gholi| 5.81| 5.81| 5.81|
|    2| bcbcbcbc| 0.58| 0.58| 0.58|
|    2| bcbcbcbc|-8.24|-8.24|-8.24|
|    2| bcbcbcbc|-0.45|-0.45|-0.45|
|    2| bcbcbcbc| 1.03| 1.03| 1.03|
|    2|aUniqueID|-0.05|-0.05|-0.05|
|    2| qqqqqqqq|-0.44|-0.44|-0.44|
+-----+---------+-----+-----+-----+
only showing top 20 rows
```

In [10]:
```
!pip install pixiedust
```

```
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes ins
tead
  from cryptography.utils import int_from_bytes
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/util.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Collecting pixiedust
  Downloading pixiedust-1.1.19.tar.gz (197 kB)
     |████████████████████████████████| 197 kB 17.6 MB/s eta 0:00:01
Collecting geojson
  Downloading geojson-2.5.0-py2.py3-none-any.whl (14 kB)
Collecting astunparse
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Requirement already satisfied: markdown in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pixiedust) (3.1.1)
Collecting colour
  Downloading colour-0.1.5-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pixiedust) (2.24.0)
Requirement already satisfied: matplotlib in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pixiedust) (3.2.2)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pixiedust) (1.0.5)
Requirement already satisfied: six<2.0,>=1.6.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from astunparse->pixiedust) (1.15.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from astunparse->pixiedust) (0.34.2)
Requirement already satisfied: setuptools>=36 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from markdown->pixiedust) (47.3.1.post20200622)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests->pixiedust) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests->pixiedust) (2.9)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests->pixiedust) (2020.12.5)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from requests->pixiedust) (1.25.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from matplotlib->pixiedust) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from matplotlib->pixiedust) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from matplotlib->pixiedust) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from matplotlib->pixiedust) (2.8.1)
Requirement already satisfied: numpy>=1.11 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from matplotlib->pixiedust) (1.18.5)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from pandas->pixiedust) (2020.1)
Building wheels for collected packages: pixiedust
  Building wheel for pixiedust (setup.py) ... done
  Created wheel for pixiedust: filename=pixiedust-1.1.19-py3-none-any.whl size=321803 sha256=ade7ce291a2c879b3e183114fb564fdb021776c54f27cd202faba94d671296cf
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/05/07/e7/8aca0e820027a63157a916424fd748fb2a2a3e71de5e08eeb8
Successfully built pixiedust
Installing collected packages: geojson, astunparse, colour, pixiedust
Successfully installed astunparse-1.6.3 colour-0.1.5 geojson-2.5.0 pixiedust-1.1.19
```

In [11]:
```
import pixiedust
display(df)
```

```
Pixiedust database opened successfully
Table VERSION_TRACKER created successfully
Table METRICS_TRACKER created successfully

Share anonymous install statistics? (opt-out instructions)

PixieDust will record metadata on its environment the next time the package is installed or updated. The data is anonymized and aggregated to help plan for future releases, a
nd records only the following values:

{
   "data_sent": currentDate,
   "runtime": "python",
   "application_version": currentPixiedustVersion,
   "space_id": nonIdentifyingUniqueId,
   "config": {
       "repository_id": "https://github.com/ibm-watson-data-lab/pixiedust",
       "target_runtimes": ["Data Science Experience"],
       "event_id": "web",
       "event_organizer": "dev-journeys"
   }
}
```

```
You can opt out by calling pixiedust.optOut() in a new cell.
```

Pixiedust version 1.1.19

```
Pixiedust runtime updated. Please restart kernel
Table SPARK_PACKAGES created successfully
Table USER_PREFERENCES created successfully
Table service_connections created successfully

DataFrame[CLASS: bigint, SENSORID: string, X: double, Y: double, Z: double]
```

In [12]:
```
df.createOrReplaceTempView("df")
```

We'll use Apache SystemML to implement Discrete Fourier Transformation. This way all computation continues to happen on the Apache Spark cluster for advanced scalability and performance.

As you've learned from the lecture, implementing Discrete Fourier Transformation in a linear algebra programming language is simple. Apache SystemML DML is such a language and as you can see the implementation is straightforward and doesn't differ too much from the mathematical definition (Just note that the sum operator has been swapped with a vector dot product using the %*% syntax borrowed from R):

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{2\pi i}{N} kn}$$
$$= \sum_{n=0}^{N-1} x_n \cdot [\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N)],$$

In [13]:
```
dml_script = '''
PI = 3.141592654
N = nrow(signal)

n = seq(0, N-1, 1)
k = seq(0, N-1, 1)

M = (n %*% t(k))*(2*PI/N)

Xa = cos(M) %*% signal
Xb = sin(M) %*% signal

DFT = cbind(Xa, Xb)
'''
```

Now it's time to create a function which takes a single row Apache Spark data frame as argument (the one containing the accelerometer measurement time series for one axis) and returns the Fourier transformation of it. In addition, we are adding an index column for later joining all axis together and renaming the columns to appropriate names. The result of this function is an Apache Spark DataFrame containing the Fourier Transformation of its input in two columns.

In [14]:
```
from pyspark.sql.functions import monotonically_increasing_id

def dft_systemml(signal,name):
    prog = dml(dml_script).input('signal', signal).output('DFT')

    return (

    #execute the script inside the SystemML engine running on top of Apache Spark
    ml.execute(prog)

         #read result from SystemML execution back as SystemML Matrix
        .get('DFT')

         #convert SystemML Matrix to ApacheSpark DataFrame
        .toDF()

         #rename default column names
        .selectExpr('C1 as %sa' % (name), 'C2 as %sb' % (name))

         #add unique ID per row for later joining
        .withColumn("id", monotonically_increasing_id())
    )
```

Now it's time to create individual DataFrames containing only a subset of the data. We filter simultaneously for accelerometer each sensor axis and one for each class. This means you'll get 6 DataFrames. Please implement this using the relational API of DataFrames or SparkSQL. Please use class 1 and 2 and not 0 and 1.

Please make sure that each DataFrame has only ONE colum (only the measurement, eg. not CLASS column)

In [16]:
```
x0 = spark.sql("SELECT X from df where class = 0") #=> Please create a DataFrame containing only measurements of class 0 from the x axis
y0 = spark.sql("SELECT Y from df where class = 0") #=> Please create a DataFrame containing only measurements of class 0 from the y axis
z0 = spark.sql("SELECT Z from df where class = 0") #=> Please create a DataFrame containing only measurements of class 0 from the z axis
x1 = spark.sql("SELECT X from df where class = 1") #=> Please create a DataFrame containing only measurements of class 1 from the x axis
y1 = spark.sql("SELECT Y from df where class = 1") #=> Please create a DataFrame containing only measurements of class 1 from the y axis
z1 = spark.sql("SELECT Z from df where class = 1") #=> Please create a DataFrame containing only measurements of class 1 from the z axis
```

Since we've created this cool DFT function before, we can just call it for each of the 6 DataFrames now. And since the result of this function call is a DataFrame again we can use the pyspark best practice in simply calling methods on it sequentially. So what we are doing is the following:

- Calling DFT for each class and accelerometer sensor axis.
- Joining them together on the ID column.
- Re-adding a column containing the class index.
- Stacking both Dataframes for each classes together

In [17]:
```
from pyspark.sql.functions import lit

df_class_0 = dft_systemml(x0,'x') \
    .join(dft_systemml(y0,'y'), on=['id'], how='inner') \
    .join(dft_systemml(z0,'z'), on=['id'], how='inner') \
    .withColumn('class', lit(0))

df_class_1 = dft_systemml(x1,'x') \
    .join(dft_systemml(y1,'y'), on=['id'], how='inner') \
    .join(dft_systemml(z1,'z'), on=['id'], how='inner') \
    .withColumn('class', lit(1))

df_dft = df_class_0.union(df_class_1)

df_dft.show()
```

```
SystemML Statistics:
Total execution time:          0.018 sec.
Number of executed Spark inst:  0.
```

```
SystemML Statistics:
Total execution time:          0.000 sec.
Number of executed Spark inst:  0.


SystemML Statistics:
Total execution time:          0.000 sec.
Number of executed Spark inst:  0.


SystemML Statistics:
Total execution time:          0.501 sec.
Number of executed Spark inst:  0.


SystemML Statistics:
Total execution time:          0.193 sec.
Number of executed Spark inst:  0.


SystemML Statistics:
Total execution time:          0.162 sec.
Number of executed Spark inst:  0.
```

id	xa	xb	ya	yb	za	zb	class
8589934592	0.0	0.0	0.0	0.0	0.0	0.0	0
0	0.0	0.0	0.0	0.0	0.0	0.0	0
8589934596	0.0	0.0	0.0	0.0	0.0	0.0	0
8589934598	0.0	0.0	0.0	0.0	0.0	0.0	0
8589934593	0.0	0.0	0.0	0.0	0.0	0.0	0
6	0.0	0.0	0.0	0.0	0.0	0.0	0
5	0.0	0.0	0.0	0.0	0.0	0.0	0
1	0.0	0.0	0.0	0.0	0.0	0.0	0
3	0.0	0.0	0.0	0.0	0.0	0.0	0
8589934594	0.0	0.0	0.0	0.0	0.0	0.0	0
8589934595	0.0	0.0	0.0	0.0	0.0	0.0	0
2	0.0	0.0	0.0	0.0	0.0	0.0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0
8589934597	0.0	0.0	0.0	0.0	0.0	0.0	0
26	0.007432298669951747	-0.00394663166701...	0.007432298669951747	-0.00394663166701...	0.007432298669951747	-0.00394663166701...	1
29	0.02589077158423112	-0.02414578651495463	0.02589077158423112	-0.02414578651495463	0.02589077158423112	-0.02414578651495463	1
8589934592	0.02589077121123453	0.024145789283483897	0.02589077121123453	0.024145789283483897	0.02589077121123453	0.024145789283483897	1
19	-0.08486126908675795	-0.0218913608396156	-0.08486126908675795	-0.0218913608396156	-0.08486126908675795	-0.0218913608396156	1
54	0.02941472472264519	0.08753524593777502	0.02941472472264519	0.08753524593777502	0.02941472472264519	0.08753524593777502	1
0	-0.00305531695825...	0.0136154100593525 76	-0.00305531695825...	0.0136154100593525 76	-0.00305531695825...	0.0136154100593525 76	1

only showing top 20 rows

Please create a VectorAssembler which consumes the newly created DFT columns and produces a column “features”

In [18]:
```
from pyspark.ml.feature import VectorAssembler
```

In [19]:
```
vectorAssembler = VectorAssembler(
    inputCols=["xa", "xb", "ya", "yb", "za", "zb"],
    outputCol="features")
```

Please insatiate a classifier from the SparkML package and assign it to the classifier variable. Make sure to set the “class” column as target.

In [23]:
```
from pyspark.ml.classification import RandomForestClassifier
```

In [24]:
```
classifier = RandomForestClassifier(labelCol="class", featuresCol="features", numTrees=10)
```

Let’s train and evaluate…

In [25]:
```
from pyspark.ml import Pipeline
pipeline = Pipeline(stages=[vectorAssembler, classifier])
```

In [26]:
```
model = pipeline.fit(df_dft)
```

```
Exception ignored in: <function JavaWrapper.__del__ at 0x7f536bd96950>
Traceback (most recent call last):
  File "/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/pyspark/ml/wrapper.py", line 40, in __del__
    if SparkContext._active_spark_context and self._java_obj is not None:
AttributeError: 'GBTClassifier' object has no attribute '_java_obj'
```

In [27]:
```
prediction = model.transform(df_dft)
```

In [28]:
```
prediction.show()
```

id	xa	xb	ya	yb	za	zb	class	features	rawPrediction	probability	prediction
8589934592	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
0	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
8589934596	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
8589934598	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
8589934593	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
8589934594	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
8589934595	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0
8589934597	0.0	0.0	0.0	0.0	0.0	0.0	0	(6,[],[])	[10.0,0.0]	[1.0,0.0]	0.0

```
0.0]|  [1.0,0.0]|       0.0|
|       26|0.007432298669951747|-0.00394663166701...|0.007432298669951747|-0.00394663166701...|0.007432298669951747|-0.00394663166701...|    1|[0.00743229866995...|   [0.0,1
0.0]|  [0.0,1.0]|       1.0|
|       29| 0.02589077158423112|-0.02414578651495463| 0.02589077158423112|-0.02414578651495463| 0.02589077158423112|-0.02414578651495463|    1|[0.02589077158423...|   [0.0,1
0.0]|  [0.0,1.0]|       1.0|
|8589934592| 0.02589077121123453|0.024145789283483897| 0.02589077121123453|0.024145789283483897| 0.02589077121123453|0.024145789283483897|    1|[0.02589077121123...|   [0.0,1
0.0]|  [0.0,1.0]|       1.0|
|       19|-0.08486126908675795| -0.0218913608396156|-0.08486126908675795| -0.0218913608396156|-0.08486126908675795| -0.0218913608396156|    1|[-0.0848612690867...|   [0.0,1
0.0]|  [0.0,1.0]|       1.0|
|       54| 0.02941472472264519| 0.08753524593777502| 0.02941472472264519| 0.08753524593777502| 0.02941472472264519| 0.08753524593777502|    1|[0.02941472472264...|   [0.0,1
0.0]|  [0.0,1.0]|       1.0|
|        0|-0.00305531695825...|0.013615410059352576|-0.00305531695825...|0.013615410059352576|-0.00305531695825...|0.013615410059352576|    1|[-0.0030553169582...|   [0.0,1
0.0]|  [0.0,1.0]|       1.0|
+---------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+-----+--------------------+---------
----+----------+----------+
only showing top 20 rows
```

In [29]:
```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
binEval = MulticlassClassificationEvaluator().setMetricName("accuracy") .setPredictionCol("prediction").setLabelCol("class")

binEval.evaluate(prediction)
```

Out[29]: 1.0

If you are happy with the result (I'm happy with > 0.8) please submit your solution to the grader by executing the following cells, please don't forget to obtain an assignment submission token (secret) from the Courera's graders web page and paste it to the "secret" variable below, including your email address you've used for Coursera.

In [30]:
```
!rm -Rf a2_m4.json
```

In [31]:
```
prediction = prediction.repartition(1)
prediction.write.json('a2_m4.json')
```

In [32]:
```
!rm -f rklib.py
!wget wget https://raw.githubusercontent.com/IBM/coursera/master/rklib.py
```

```
--2021-04-16 20:08:03--  http://wget/
Resolving wget (wget)... failed: Name or service not known.
wget: unable to resolve host address 'wget'
--2021-04-16 20:08:03--  https://raw.githubusercontent.com/IBM/coursera/master/rklib.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.111.133, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2540 (2.5K) [text/plain]
Saving to: 'rklib.py'

rklib.py            100%[===================>]   2.48K  --.-KB/s    in 0s

2021-04-16 20:08:04 (30.4 MB/s) - 'rklib.py' saved [2540/2540]

FINISHED --2021-04-16 20:08:04--
Total wall clock time: 0.3s
Downloaded: 1 files, 2.5K in 0s (30.4 MB/s)
```

In [33]:
```
from rklib import zipit
zipit('a2_m4.json.zip','a2_m4.json')
```

In [34]:
```
!base64 a2_m4.json.zip > a2_m4.json.zip.base64
```

In [36]:
```
from rklib import submit
key = "-fBiYHYDEeiR4QqiFhAvkA"
part = "IjtJk"
email = "tjamesbu@gmail.com"
submission_token = "wTowrJg2oO8XWuD0" # (have a look here if you need more information on how to obtain the token https://youtu.be/GcDo0Rwe06U?t=276)

with open('a2_m4.json.zip.base64', 'r') as myfile:
    data=myfile.read()
submit(email, submission_token, key, part, [part], data)
```

```
Submission successful, please check on the coursera grader page for the status
------------------------
{"elements":[{"itemId":"B8wXV","id":"f_F-qCtuEei_fRLwaVDk3g~B8wXV~fBPZR57vEeu7agq7wAioGw","courseId":"f_F-qCtuEei_fRLwaVDk3g"}],"paging":{},"linked":{}}
------------------------
```

In [ ]:
```
```