Welcome to exercise one of week three of “Apache Spark for Scalable Machine Learning on BigData”. In this exercise we’ll use the HMP dataset again and perform some basic operations using Apache SparkML Pipeline components.

Let’s create our DataFrame again:

In [ ]:
```
# delete files from previous runs
!rm -f hmp.parquet*

# download the file containing the data in PARQUET format
!wget https://github.com/IBM/coursera/raw/master/hmp.parquet

# create a dataframe out of it
df = spark.read.parquet('hmp.parquet')

# register a corresponding query table
df.createOrReplaceTempView('df')
```

Given below is the feature engineering pipeline from the lecture. Please add a feature column called “features_minmax” using the MinMaxScaler.

More information can be found here: http://spark.apache.org/docs/latest/ml-features.html#minmaxscaler

In [ ]:
```
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorAss
embler, Normalizer, MinMaxScaler
from pyspark.ml.linalg import Vectors
from pyspark.ml import Pipeline

indexer = StringIndexer(inputCol="class", outputCol="classIndex")
encoder = OneHotEncoder(inputCol="classIndex", outputCol="categoryVec"
)
vectorAssembler = VectorAssembler(inputCols=["x","y","z"],
                                  outputCol="features")
normalizer = Normalizer(inputCol="features", outputCol="features_norm"
, p=1.0)

minmaxscaler = $$

pipeline = Pipeline(stages=[indexer, encoder, vectorAssembler, normali
zer,minmaxscaler])
model = pipeline.fit(df)
prediction = model.transform(df)
prediction.show()
```

The difference between a transformer and an estimator is state. A transformer is stateless whereas an estimator keeps state. Therefore “VectorAsselmbler” is a transformer since it only need to read row by row. Normalizer, on the other hand need to compute statistics on the dataset betfore, therefore it is an estimator. An estimator has an additional “fit” function. “OneHotEncoder” has been deprecated in Spark 2.3, therefore please change the code below to use the OneHotEstimator instead of the “OneHotEncoder”.

More information can be found here: http://spark.apache.org/docs/latest/ml-features.html#onehotencoderestimator

In [ ]:

```
from pyspark.ml.feature import OneHotEncoder, StringIndexer, VectorAss
embler, Normalizer, MinMaxScaler, OneHotEncoderEstimator
from pyspark.ml.linalg import Vectors
from pyspark.ml import Pipeline

indexer = StringIndexer(inputCol="class", outputCol="classIndex")
encoder = OneHotEncoder(inputCol="classIndex", outputCol="categoryVec"
)
vectorAssembler = VectorAssembler(inputCols=["x","y","z"],
                                  outputCol="features")
normalizer = Normalizer(inputCol="features", outputCol="features_norm"
, p=1.0)


pipeline = Pipeline(stages=[indexer, encoder, vectorAssembler, normali
zer])
model = pipeline.fit(df)
prediction = model.transform(df)
prediction.show()
```