Task 3: Skeleton of an ML program: Transformer, Estimator, Parameters

Objective: Understand the building block of any ML application with the example of predicting the role of an IT professional is developer by looking at his experience and annual salary

```
#Read data
df = spark.read.csv("/home/s_kante/spark/data/developers_survey_training.csv", header='true')

#Replace IsDeveloper value with integer 1 or 0
df.createOrReplaceTempView("inputData")
df1 = spark.sql("SELECT CASE IsDeveloper WHEN 'Yes' THEN 1 ELSE 0 END AS label,
CAST(YearsOfExp AS FLOAT) AS YearsOfExp, CAST(Salary AS FLOAT) AS Salary FROM
inputData ");

#Create feature vector
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=["Salary","YearsOfExp"], outputCol="features")
combined = assembler.transform(df1)
vector_df = combined.select(combined.label, combined.features)

#Estimator: Create an instance LogisticRegression which is an estimator
from pyspark.ml.classification import LogisticRegression
lr_estimator = LogisticRegression(maxIter=10)
print str(LogisticRegression().explainParams())

#Train the model
model = lr_estimator.fit(vector_df)

#Parameters: Check the parameters used to train the model
params = model.extractParamMap()

#Pass parameters explicitly while training the model
params = {lr_estimator.maxIter:15}
model = lr_estimator.fit(vector_df, params)

#Transformer: test the model. Transform method will return a dataframe with predictions
prediction = model.transform(vector_df)

#Save the model on disc
model.save("/home/s_kante/spark/data/trained_models/predict_emp_role")

#Load a trained model from disc to memory
from pyspark.ml.classification import LogisticRegressionModel
```

```
mymodel =
LogisticRegressionModel.load("/home/s_kante/spark/data/trained_models/predict_emp_role")
prediction = mymodel.transform(vector_df)
```

#QUESTION: How do we predict in actual production environment?