**Task 6: K-means**

**Objective:** K-means clustering an instance of Unsupervised ML

#Read data

df = spark.read.csv("/home/s\_kante/spark/data/Task6/Wine.csv", header='true')

#Replace IsDeveloper value with integer 1 or 0

df1 = df.select(df.Alcohol.cast("float"), df.Malic\_Acid.cast("float"), df.Ash.cast("float"), df.Ash\_Alcanity.cast("float"), df.Magnesium.cast("float"), df.Total\_Phenols.cast("float"), df.Flavanoids.cast("float"), df.Nonflavanoid\_Phenols.cast("float"), df.Proanthocyanins.cast("float"), df.Color\_Intensity.cast("float"), df.Hue.cast("float"), df.OD280.cast("float"), df.Proline.cast("float"))

#Create feature vector

from pyspark.ml.feature import VectorAssembler

assembler = VectorAssembler(inputCols=["Alcohol", "Malic\_Acid", "Ash", "Ash\_Alcanity", "Magnesium", "Total\_Phenols", "Flavanoids", "Nonflavanoid\_Phenols", "Proanthocyanins", "Color\_Intensity", "Hue", "OD280", "Proline"], outputCol="features")

combined = assembler.transform(df1)

vector\_df = combined.select(combined.features)

#Let the algorithm figure out different clusters

from pyspark.ml.clustering import KMeans

kmeans = KMeans().setK(3)

model = kmeans.fit(vector\_df)

#Predict

predict = model.transform(vector\_df)

predict.show()