Big Data Analytics

Dr Sirintra Vaiwsri | Email: sirintra.v@itm.kmutnb.ac.th

Clustering

Clustering (Dietrich et al., 2015; Guller, 2015)

- Clustering is one of the unsupervised learning algorithms where the labels will not be determined to apply to the clusters.
- Therefore, clustering is used with unlabeled datasets.
- Clustering finds the similarities between objects based on the object attributes and groups the similar objects into clusters.
- The dataset is split into clusters, where elements in the same cluster are more similar to each other than elements in the other clusters.
- Clustering tasks in unsupervised learning algorithms are such as:
  - K-means
  - Principal Component Analysis (PCA)
  - Singular Value Decomposition (SVD)

K-Mean (Chambers and Zaharia, 2018; UC, 2023)

- K-Mean is one of the clustering algorithms.
- It is usually used with numerical data.
- It is used for grouping data points into *k* number of groups.
- The data points in the same group are similar, whereas they are dissimilar to data points in the other groups.

K-Mean (Chambers and Zaharia, 2018; UC, 2023)

- Steps:
  - Initialise *k* clusters using random *k* points (called the centroids)
  - For each new data point, find the distance to the centroid (Euclidean distance), then assign the point to the closest cluster.
  - Update the cluster centroid by calculating mean values.
  - Repeat steps until cluster centroids are not changed or until they reach the maximum number of iterations.

K-Mean Evaluation Example

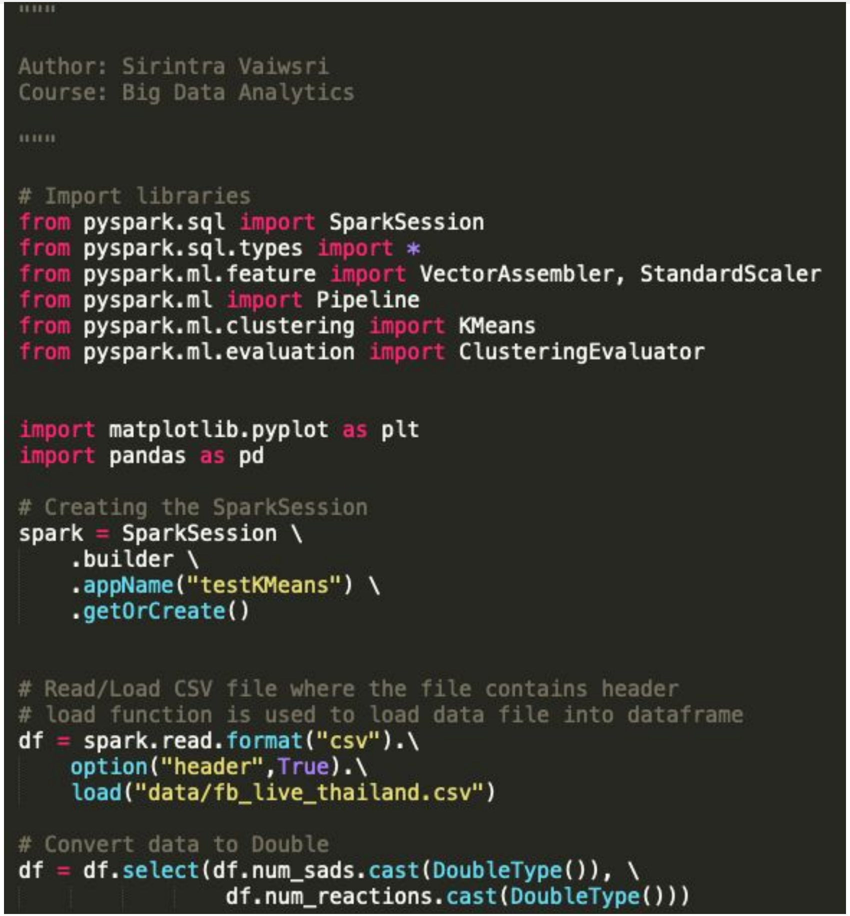
(Scikit-learn, 2023; UC, 2023)

- Silhouette analysis is used to measure the quality of clusters.
- It indicates how far data is in clusters.
- The measure is in the range of [-1, 1]:
    - -1 means the data might be assigned to the wrong cluster.
    - 0 means data in the clusters are very close.
    - 1 means data in the clusters are far away from each other.
- The *k* number that provides the highest average of the silhouette is the best *k* for the given data.

K-Mean Implementation Example

K-Mean Implementation

1. Import libraries
2. Create SparkSession
3. Load data file into Dataframe
4. Convert data to Double type



```
"""

Author: Sirintra Vaiwsri
Course: Big Data Analytics

"""

# Import libraries
from pyspark.sql import SparkSession
from pyspark.sql.types import *
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.ml import Pipeline
from pyspark.ml.clustering import KMeans
from pyspark.ml.evaluation import ClusteringEvaluator


import matplotlib.pyplot as plt
import pandas as pd

# Creating the SparkSession
spark = SparkSession \
    .builder \
    .appName("testKMeans") \
    .getOrCreate()


# Read/Load CSV file where the file contains header
# load function is used to load data file into dataframe
df = spark.read.format("csv").\
    option("header",True).\
    load("data/fb_live_thailand.csv")

# Convert data to Double
df = df.select(df.num_sads.cast(DoubleType()), \
                df.num_reactions.cast(DoubleType()))
```

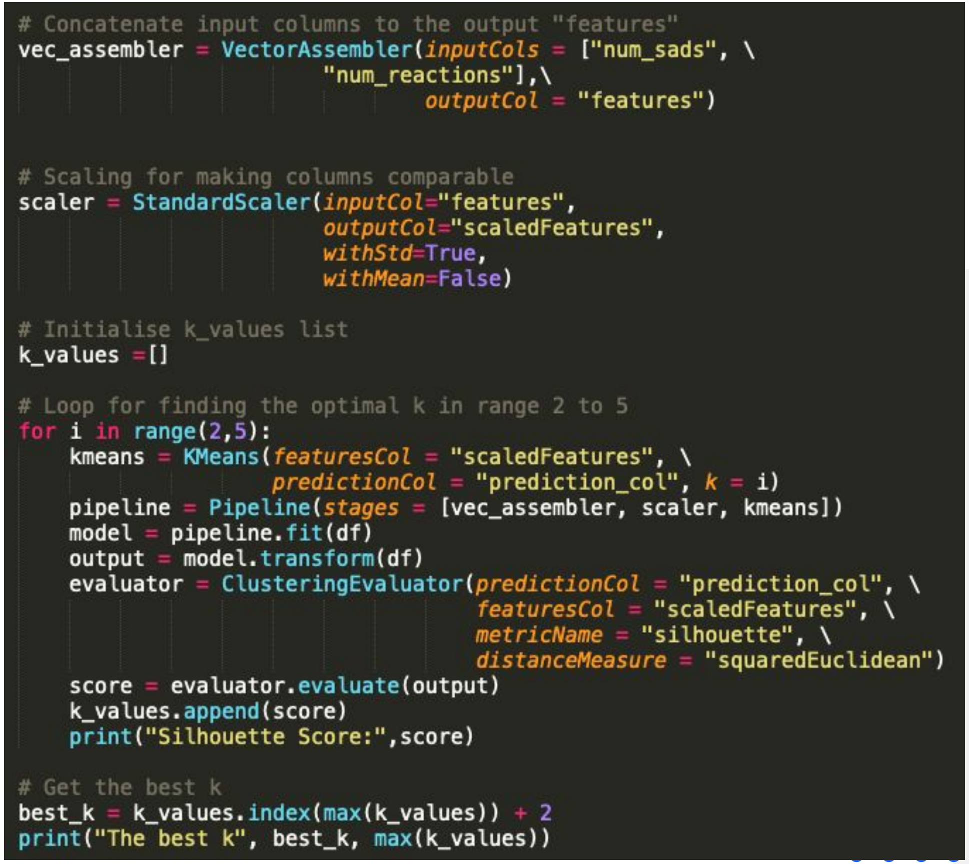
K-Mean Implementation

5. Prepare vector for features

6. Scale data to make them comparable

7. Loop for finding the best *k* number

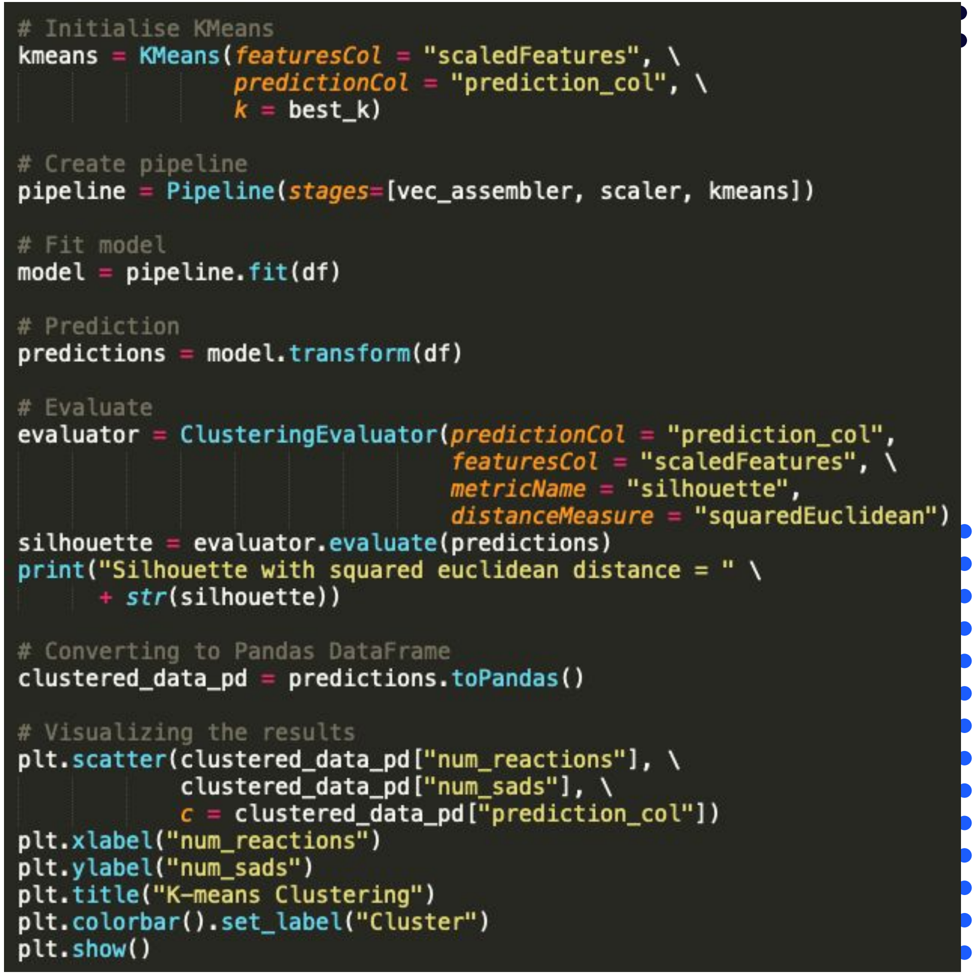
8. Get the best *k*



```
# Concatenate input columns to the output "features"
vec_assembler = VectorAssembler(inputCols = ["num_sads", \
                        "num_reactions"],\
                                outputCol = "features")


# Scaling for making columns comparable
scaler = StandardScaler(inputCol="features",
                        outputCol="scaledFeatures",
                        withStd=True,
                        withMean=False)

# Initialise k_values list
k_values =[]

# Loop for finding the optimal k in range 2 to 5
for i in range(2,5):
    kmeans = KMeans(featuresCol = "scaledFeatures", \
                    predictionCol = "prediction_col", k = i)
    pipeline = Pipeline(stages = [vec_assembler, scaler, kmeans])
    model = pipeline.fit(df)
    output = model.transform(df)
    evaluator = ClusteringEvaluator(predictionCol = "prediction_col", \
                                    featuresCol = "scaledFeatures", \
                                    metricName = "silhouette", \
                                    distanceMeasure = "squaredEuclidean")
    score = evaluator.evaluate(output)
    k_values.append(score)
    print("Silhouette Score:",score)

# Get the best k
best_k = k_values.index(max(k_values)) + 2
print("The best k", best_k, max(k_values))
```

K-Mean Implementation

9. Initialise KMeans
10. Create a pipeline
11. Fit data to model
12. Transform
13. Evaluate
14. Visualise



```
# Initialise KMeans
kmeans = KMeans(featuresCol = "scaledFeatures", \
                predictionCol = "prediction_col", \
                k = best_k)

# Create pipeline
pipeline = Pipeline(stages=[vec_assembler, scaler, kmeans])

# Fit model
model = pipeline.fit(df)

# Prediction
predictions = model.transform(df)

# Evaluate
evaluator = ClusteringEvaluator(predictionCol = "prediction_col",
                                featuresCol = "scaledFeatures", \
                                metricName = "silhouette",
                                distanceMeasure = "squaredEuclidean")
silhouette = evaluator.evaluate(predictions)
print("Silhouette with squared euclidean distance = " \
      + str(silhouette))

# Converting to Pandas DataFrame
clustered_data_pd = predictions.toPandas()

# Visualizing the results
plt.scatter(clustered_data_pd["num_reactions"], \
            clustered_data_pd["num_sads"], \
            c = clustered_data_pd["prediction_col"])
plt.xlabel("num_reactions")
plt.ylabel("num_sads")
plt.title("K-means Clustering")
plt.colorbar().set_label("Cluster")
plt.show()
```

Assignment (1 point)

- Please implement the code in slides 8 to 10.
- Please execute your code and show the result to get 1 point.

References

- Dietrich, D., Heller, B., & Yang, B. (2015). *Data science & big data analytics: discovering, analyzing, visualizing and presenting data*. Wiley.
- Guller, M. (2015). Big data analytics with spark.
- Chambers, B., & Zaharia, M. (2018). Spark: The definitive guide: Big data processing made simple. " O'Reilly Media, Inc.".
- UC. University of Cincinnati. https://uc-r.github.io/kmeans_clustering. Accessed: 2023-09-14.
- Scikit-learn. https://scikit-learn.org. Accessed: 2023-09-14.