**Algorithm**(processed\_dataset, support\_threshold){

       interesting\_itemsets ← Null

       data\_frames ← empty dictionary

       kmeans\_results ← empty dictionary

       interesting\_rules ← empty list

       categorical\_subdataset ← processed\_dataset.categorical\_subdataset

       encoded\_data ← encode(categorical\_subdataset)

       frequent\_itemsets ← fp-growth(encoded\_data,  support\_threshold)

       interesting\_itemsets ← frequent\_itemsets  with length >= 2

       item\_set ← 0

     While item\_set < interesting\_itemsets.length {

             sub\_df ← sub-data-frame of rows containing all items in the item\_set

             sub\_df\_name ← name the sub-data-frame based on the item\_set

             data\_frames[sub\_df\_name] ← sub\_df\_name

       }

       for each key in data\_frames{

             drop the column of the quantitative value meant on the right side of the rules

             k\_encoded ← encode(data\_frames.categorical-columns)

             silhouette\_best ← -1

             param\_grid ← {‘n-clusters’: [list of k values]}

             k\_best ← param\_grid[‘n-clusters’][0]

             for each cluster\_count in param\_grid{

                   k\_means ← K-Means(cluster\_count)

                   silhouette\_avg ← silhouette-score(k\_encoded, k\_means)

                   if silhouette\_avg > silhouette\_best{

                         silhouette\_best ← silhouette\_avg

                         k\_best ← cluster\_count

                   }

             }

             k\_means ← K-Means(k\_best)

             Assign labels to rows using k\_means.labels

             right\_side\_avg ← average of the quantitative column on the right side of the rules

             for each numeric column per cluster{

                   Calculate their min and max

             }

             results\_by\_label ← empty list

             i ← 0

             for each label{

                   rule ← Summary string including all the ranges for the numeric columns and right\_side\_avg

                   results\_by\_label[i] ← rule

                   Increase i

             }

             kmeans\_results[key] ← results\_by\_label

                i ← 0

             for each rule in kmeans\_results{

                   if the rule passes a z-test{

                         interesting\_rules[i] ← rule

                   }

             }

       }

       return interesting\_rules

}