DATA EXPLORATION USING CLUSTERS

Abstract

Implementation and evaluation of clustering methods to enable faster exploratory data analysis

Prasanna Iyer

Contents

High-level overview

Data in the business world is growing exponentially. According to IDC report "The Digitization of the World: From Edge to Core", 79 zeta bytes of data was generated in 2021. Most data are generated by operational activities, such as customer transactions, supply chain management, and product development. This is also reflected in the prominence of services to manage and analyze data. This is not at all surprising. Business professionals experience this every day. There is a steady accumulation of data pertaining to shipments, invoices, customers, inventory, people etc.

The challenge is specifically gaining insight from this large amount of data. There are tools such as Tableau or PowerBI that are suitable for data visualization and creation of appealing charts. However, these tools are helpful when one knows what to look for in the data.

A particularly appealing approach that could speed up the understanding of a large dataset is segmentation. This technique refers to the segmentation of a dataset into groups based on similarity. Segmentation through visual inspection is feasible with low dimensional data [up to 3 dimensions]. Beyond 3 dimensions, a more systematic approach is necessary to utilize the information from the available attributes.

Each segment (or grouping of data) can be considered as a particular type of input data represented in the dataset and can therefore enable deeper understanding. For instance, segmentation can be used to identify:

1. Key customer types from a dataset of customer transactions [e.g. 5 customer groups from 500 customers]
2. Main cylinder profiles from part master data for cylinder commodity [e.g. 10 main cylinder profiles from 600 SKUs]
3. Major types of shipment lanes from historic shipment data [50 main lanes from 10,000 monthly shipments]

In the above examples, the main benefit is that it is easier to review 5 customer groups [vs. 500 individual customers], 10 cylinder profiles [vs. 600 SKUs] or 50 main shipment lanes [vs. 10,000 shipments] and thereby, gain a deeper understanding of the data.

The purpose of this project is to demonstrate the steps for data segmentation.

Description of input data

This case study covers a scenario overlapping between engineering and procurement. The dataset contains synthetic information about a homogenous group of part numbers. Such a group of parts is also referred to as commodity. Using the dataset, the case study demonstrates the approach for data segmentation.

Each line item in the data represents combination of a particular SKU [Stock Keeping Unit] and using location. From a procurement perspective, the objective is to gain a deep understanding of the types of products. This involves exploring the commonalities as well as the differences among the SKUs. Such an understanding will enable procurement specialists to identify and evaluate the key supplier capabilities for the segments.

The attributes in the dataset can be categorized as shown in figure 1.

Category	Description	Attributes
1	Geometrical: These attributes specify the physical aspects of the SKU	Attr1_Str_qu, Attr4_RdDi_norm, Attr5_BrDi_norm
2	Cost: Refers to the cost of the SKU	Attr3_Costamount_norm
3	Technical: These attributes specify technical aspects about the interface between the SKU and the overall application	Attr2_OpPr_qu, Attr5_Str_Type1, Attr5_Str_Type2, Attr5_Str_Type3, Attr6_Major_Type1, Attr6_Major_Type2, Attr7_Cshn1, Attr7_Cshn2, Attr7_Cshn3, Attr7_Cshn4, Attr8_Repha1, Attr8_Repha2, Attr8_Repha3, Attr9_Frctn0, Attr9_Frctn1, Attr10_Snsr0, Attr10_Snsr1
4	Business: These attributes contain region and business unit information	p_region, p_bu

Figure 1: Description of input data

Strategy for solving the problem

The objective of the project is to segment the input data. Input data contains several attributes about the SKUs and therefore, a manual approach for data segmentation will not be feasible. Clustering algorithm suitable for medium dimensional data would be utilized.

One of the first steps would be the Exploratory Data Analysis (EDA) of the input dataset. Based on the EDA, attributes (to be used in the clustering) would be selected.

Dataset containing the selected attributes would be utilized by the clustering algorithm. The next step would be to derive useful insights about each of the data segments.

Discussion of the expected solution

Clustering would be expected to yield data segments (or clusters) that can be shown to be well separated. The resulting data segments would need to be well defined such that for each segment, the distance between points within the segment should be less than the distance to points in other segments. This would ensure that the points within a segment are similar to each other and there are meaningful differences between the segments.

Metrics with justification

The expected solution should deliver segments that are well separated from each other. The estimation of similarity between points is a critical step. The quality of the clustering can be measured by comparing the distance between points within a cluster versus the distance between points assigned to different clusters. As shown in figure 13, there are 3 metrics utilized to evaluate the quality of clusters from each experiment.

1) Silhouette score[1]: The silhouette score is composed of two scores –

   - **a**: The mean distance between a sample and all other points in the same class
   - **b**: The mean distance between a sample and all other points in the *next nearest cluster*

   $$s = \frac{b - a}{\max(a, b)}$$

   Silhouette score is an appropriate option for two reasons. Firstly, the Silhouette score compares the closeness of the points in a class versus the distance between classes and this approach meets the project requirement. Secondly, Silhouette score is a suitable option when ground truth labels are not known (as is the case with this project)

2) Calinski Harabasz Index[2]: The Calinski-Harabasz score is defined as the ratio of the between-clusters dispersion mean and the within-cluster dispersion. Thus, this score measures the closeness of the points within a cluster and the separation between the clusters. Also, this score works well when the ground truth labels are not available.

3) Davies Bouldin score[3]: The score is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters which are farther apart and less dispersed will result in a better score. Also, this score works well when the ground truth labels are not available.

The three metrics use different approaches to basically compare the separation between the clusters versus the proximity of the points within a cluster. These three metrics also work for use cases where the ground truth label is not available

Metric number	Name	Description
1	Silhouette score	Best value is 1 and worst value is -1
2	Calinski Harabasz	Higher value is better
3	Davies Bouldin	Lower value is better

Figure 2: Cluster evaluation metrics

In this scenario, several scenarios will be modeled and the parameter setting that yields the top performance across the 3 metrics will be selected.

Exploratory data analysis

The dataset contains 648 rows and 28 columns [features]. Among the features, 21 are numeric and 5 are alpha-numeric. The numeric features [excluding Attr3_Costamount_norm] contain technical specifications of the SKUs. The dataset has been generalized and therefore, the attributes do not reflect real values.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz_score.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html

Each SKU in the dataset belongs to a particular region and business unit. Figure 1 shows the stratification of SKUs by region. Region_1 has the highest number of SKUs (at 48% of total) followed by Region_2 (at 31% of total).

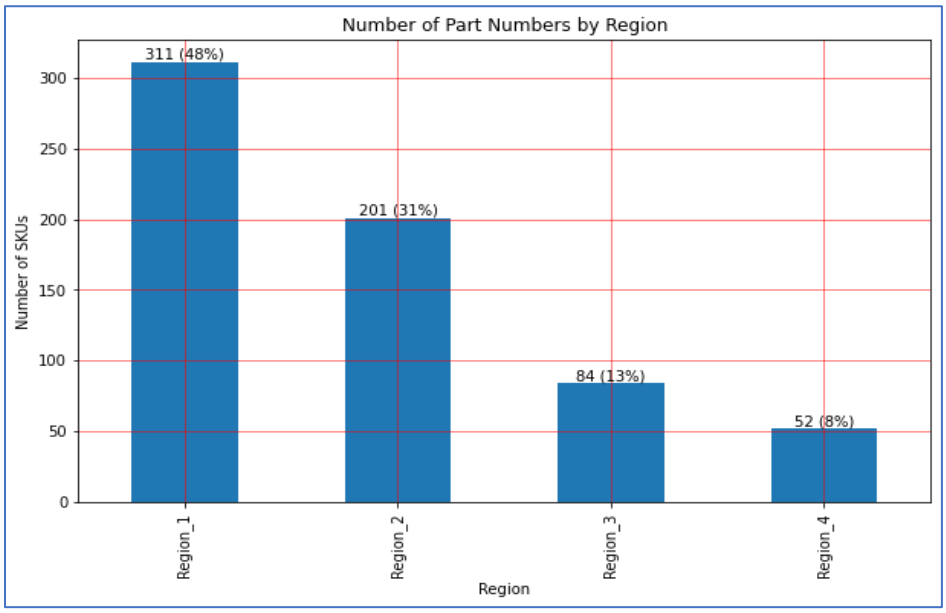

Figure 3: Number of SKUs by Region

Figure 3 shows the stratification of SKUs by business unit. “AG” business unit (p_bu = “AG”) has the highest number of SKUs (44%) followed by “AS” at 30%. Based on the significance of the business units, this case study will focus on “AG” and “CE” business units.

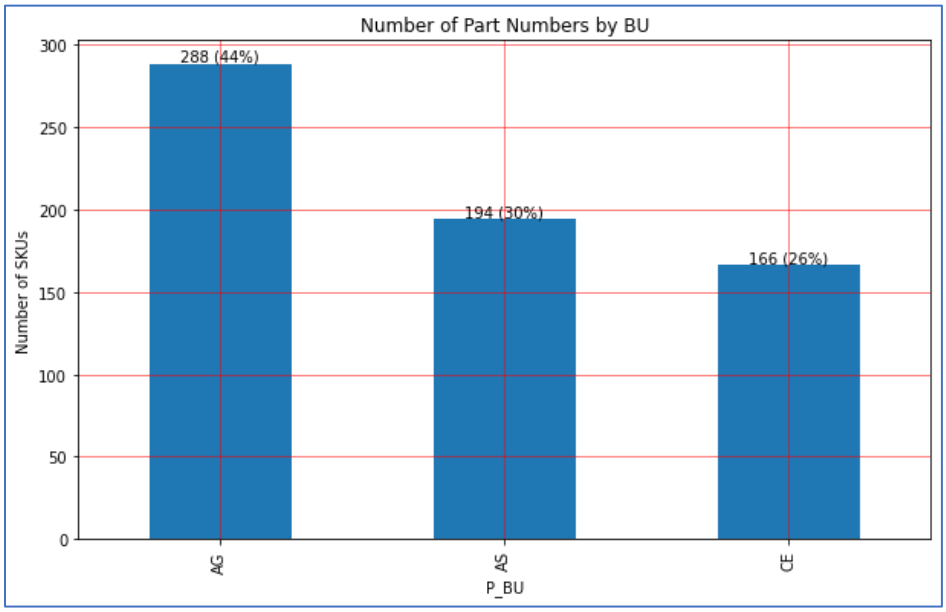

Figure 3: Number of SKUs by Business Unit

Figure 4 shows the average cost by region. Average SKU_cost for Region_1 is higher than the other regions. Region_2 and Region_4 have similar average costs.

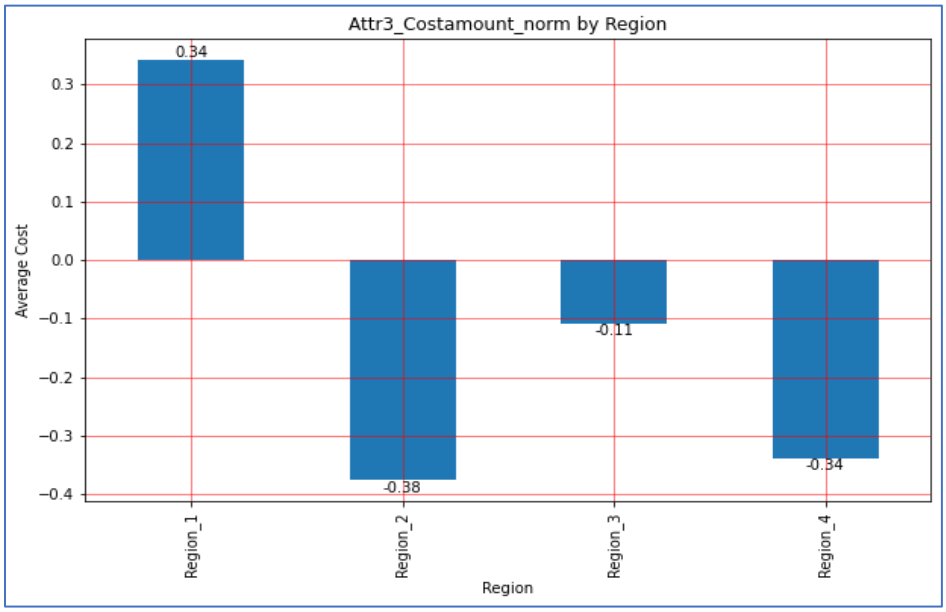

Figure 4: Average SKU Cost by Region

Correlation

One of the first steps is to understand the features whose values have an influence on the cost of the SKU. Covariance among the attributes and specifically versus cost, can be used to identify the features that are correlated with cost.

Figure 4 shows the correlation values among the features. Attr4 and Attr5 show the highest correlation with cost. This is intuitive since Attr4_RdDi_norm and Attr5_BrDi_norm describe the geometry of the SKU. Geometry is one of the main drivers of the raw material needed to manufacture and therefore, impact the cost.

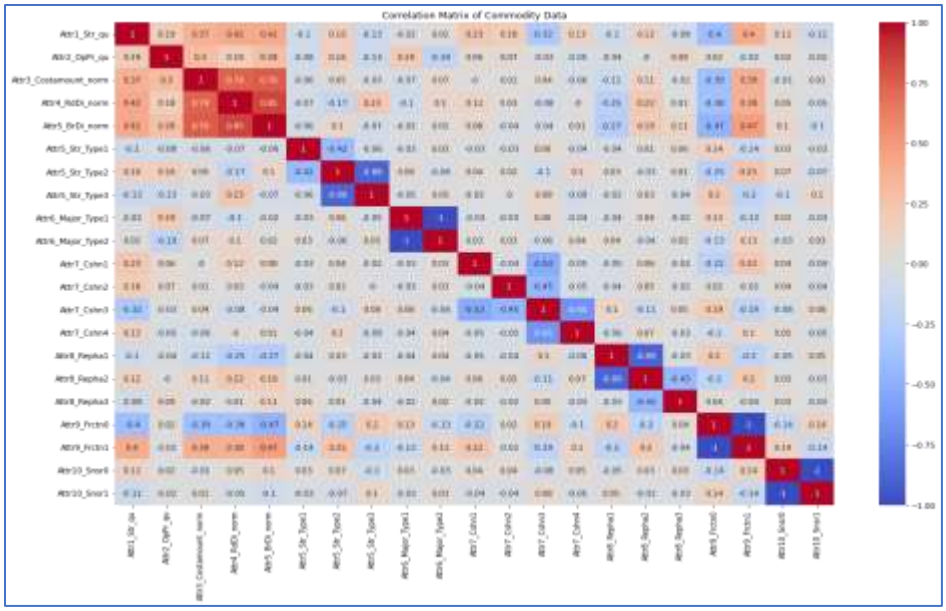

Figure 4: Correlation heatmap for all numeric features

Figure 5 shows the correlation heatmap for a subset of features with correlation value (versus Attr3_Costamount_norm) greater than 0.3.

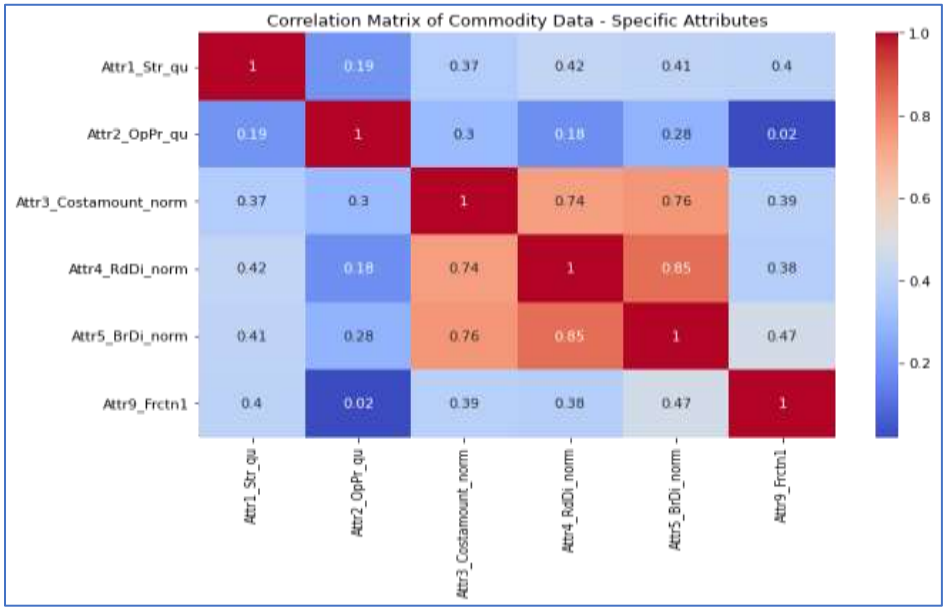

Figure 5: Correlation heatmap for feature subset

Attr1_Str_qu and Attr9_Frct1 also show meaningful correlation with cost. The influence of Attr1_Str_qu on the cost is understandable as it partially describes the geometry. Attr9_Frctn1 refers to a specialized welding requirement and therefore, the impact on cost is as expected.

Correlation by Region

The SKUs in the dataset belong to 4 different regions. The list of features with highest correlation versus cost varies by region:

1. Region 1, Region 3, Region 4: Attr5_BrDi_norm [related to geometry] shows the highest correlation [range: 0.71 – 0.91] with cost
2. Region 2: Attr1_Str_qu [related to geometry] shows the highest correlation [0.56] with cost

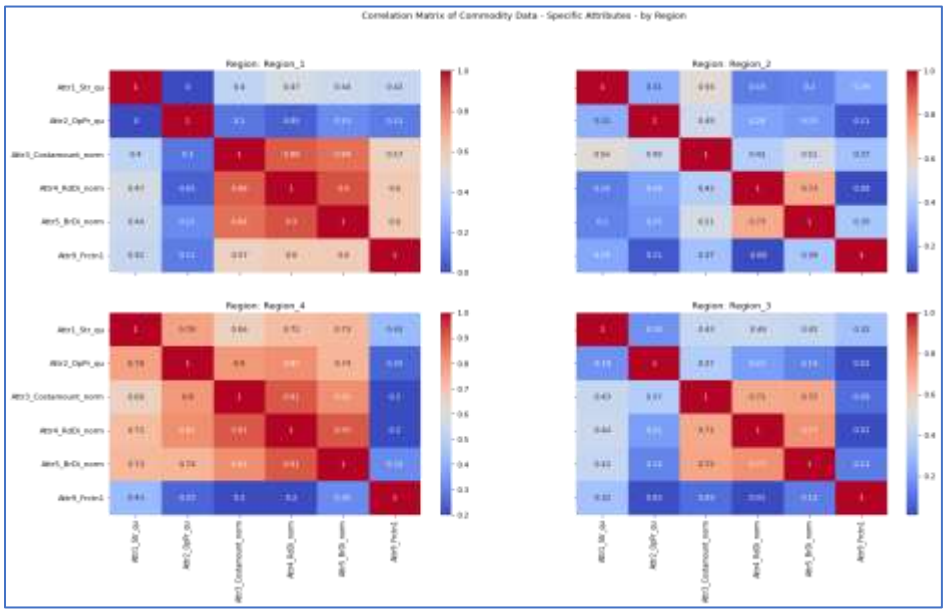

Figure 6: Correlation heatmap by region

Correlation by Business Unit

The SKUs in the dataset belong to 2 main business units [AG & CE]. Figure 7 shows the correlation values for each business unit.

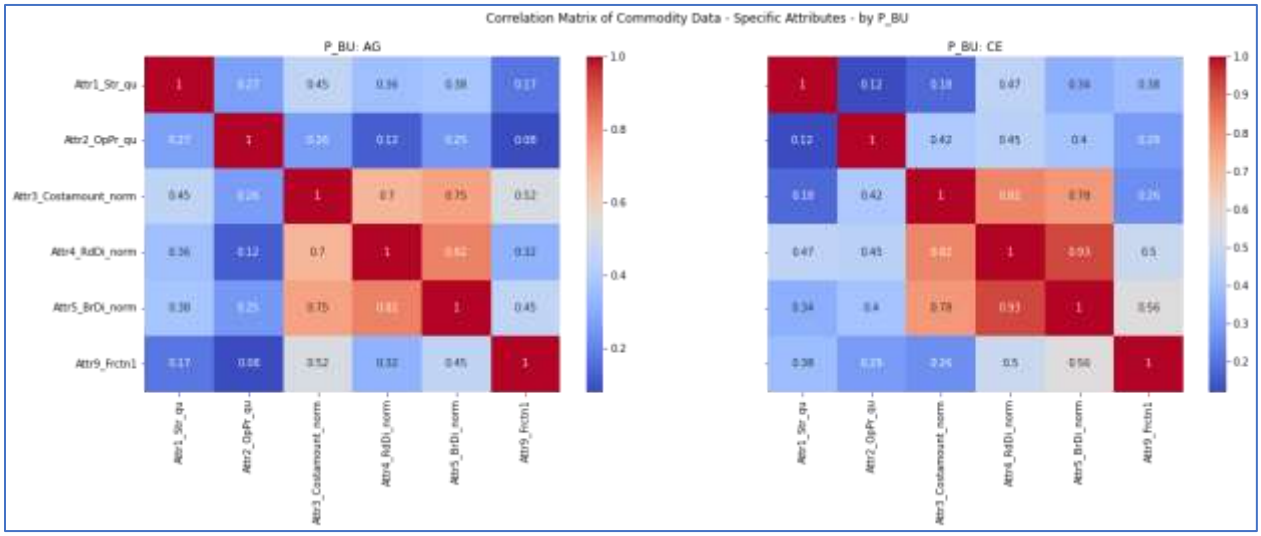

Figure 7: Correlation by Business Unit

For the “AG” business unit, Attr4_RdDi_norm and Attr5_BrDi_norm show the highest correlation with cost. In addition, Attr9_Frctn1 and Attr1_Str_qu also show meaningful correlations of 0.52 and 0.45 respectively.

For the “CE” business unit, Attr4_RdDi_norm and Attr5_BrDi_norm also show the highest correlation with cost. However, the correlation values for Attr4_RdDi_norm and Attr5_BrDi_norm for the CE business unit is greater than the AG business unit. The variance in Attr9_Frct1 within the “CE” business unit is minimal (majority of the SKUs have this special welding requirement) and therefore, lack of correlation with Atr3_Cost_norm is as expected.

Distribution of Specific Attributes

In this section, the distribution of the attributes that exhibited meaningful correlation with Attr3_Cost_norm has been reviewed. The two attributes whose distribution (within each business unit) will be studied are Attr1_Str_qu and Attr2_OpPr_qu

Distribution of Attr1_Str_qu by Business Unit

Figure 8 shows the distribution of Attr1_Str_qu by Business Unit. From figure 8, it can be observed that the variance for Attr1_Str_qu for “AG” is significantly greater than “CE”. This is also reflected in the variance coefficient [AG = 0.69, CE = 0.38]. Therefore, it can be concluded that Attr1_Str_qu has a higher influence over Attr3_Cost_norm for the “AG” business unit.

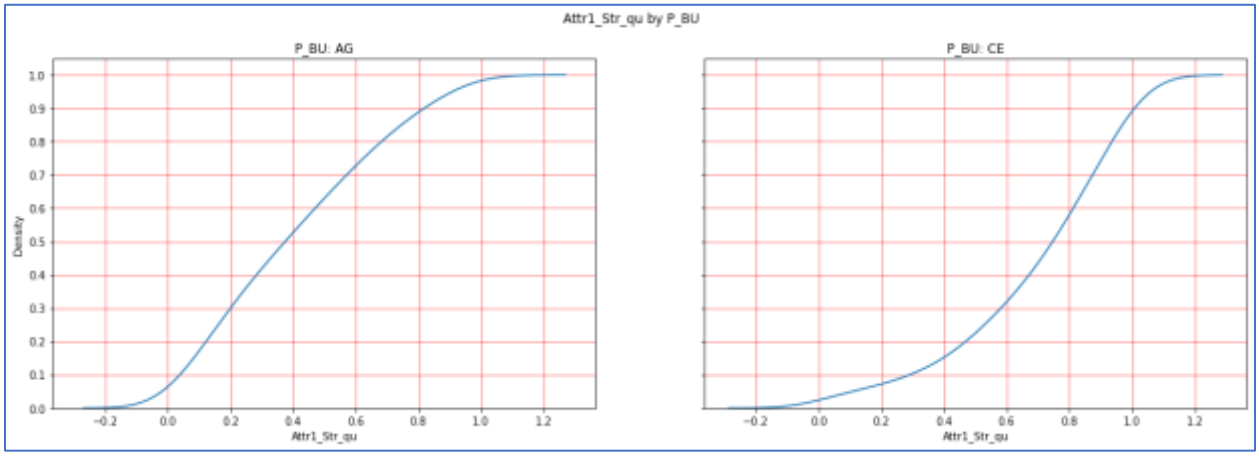

Figure 8: Distribution of Attr1_Str_qu by Business Unit

Distribution of Attr2_OpPr_qu by Business Unit

Figure 9 shows the distribution of Attr2_OpPr_qu by business unit. From figure 9, it can be observed that the variance for Attr2_OpPr_qu is greater for “AG” business unit compared to “CE” business unit. This is also reflected in the variance coefficients. The variance coefficient for “AG” is 41% greater than “CE” (AG = 0.65, CE = 0.46).

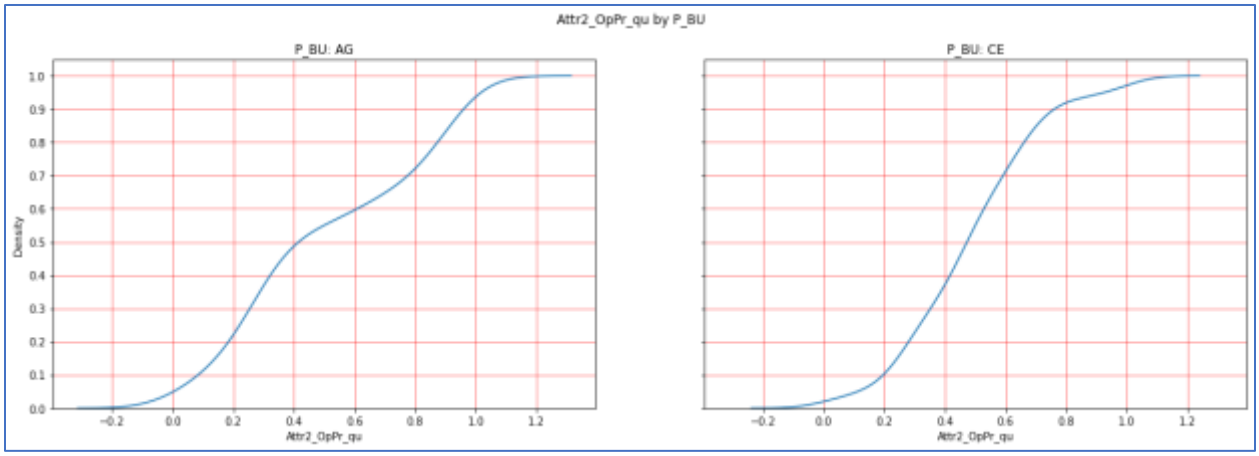

Figure 9: Distribution of Attr2_OpPr_qu by Business Unit

Data preprocessing

In this step, subset of attributes based on variance are selected. The list below shows the 7 main attributes to be included in the model building step.

```
['Attr1_Str_qu', 'Attr2_OpPr_qu', 'Attr3_Costamount_norm',
       'Attr4_RdDi_norm', 'Attr5_BrDi_norm', 'Attr9_Frctn0', 'Attr9_Frctn1']
```

Abnormalities in the input data

Based on the specific distribution of the numeric features in the input data appropriate pre-processing. Figure XX shows the distribution of the original costamount attribute. The distribution is nearly normal. Therefore, as part of pre-processing, this attribute will be normalized:

$$x_{normalized} = \frac{x - x.mean}{x.std\_dev}$$

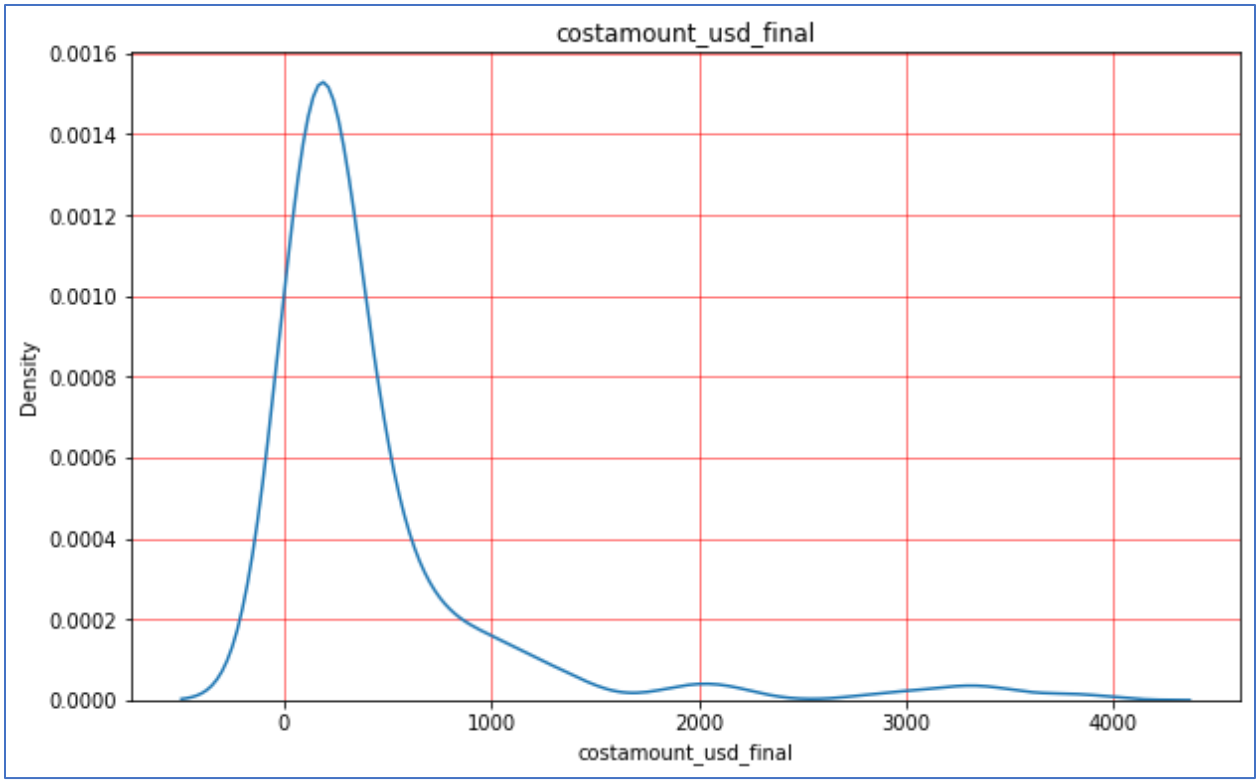

Figure 4: Distribution of Attr3_costamount

Similarly figure XX shows the distribution of the original Attr2_OpPr. The distribution shows multiple peaks and does not follow the normal distribution. In this instance, quantiles will be used in the pre-processing[4]:

1) Pick the number of quantiles
2) Split the data into quantiles, with each quantile containing equal number of examples
3) Replace each example by the index of the quantile it falls in
4) Scale the index values (using min – max scaling) to [0, 1]

[4] https://developers.google.com/machine-learning/clustering/prepare-data

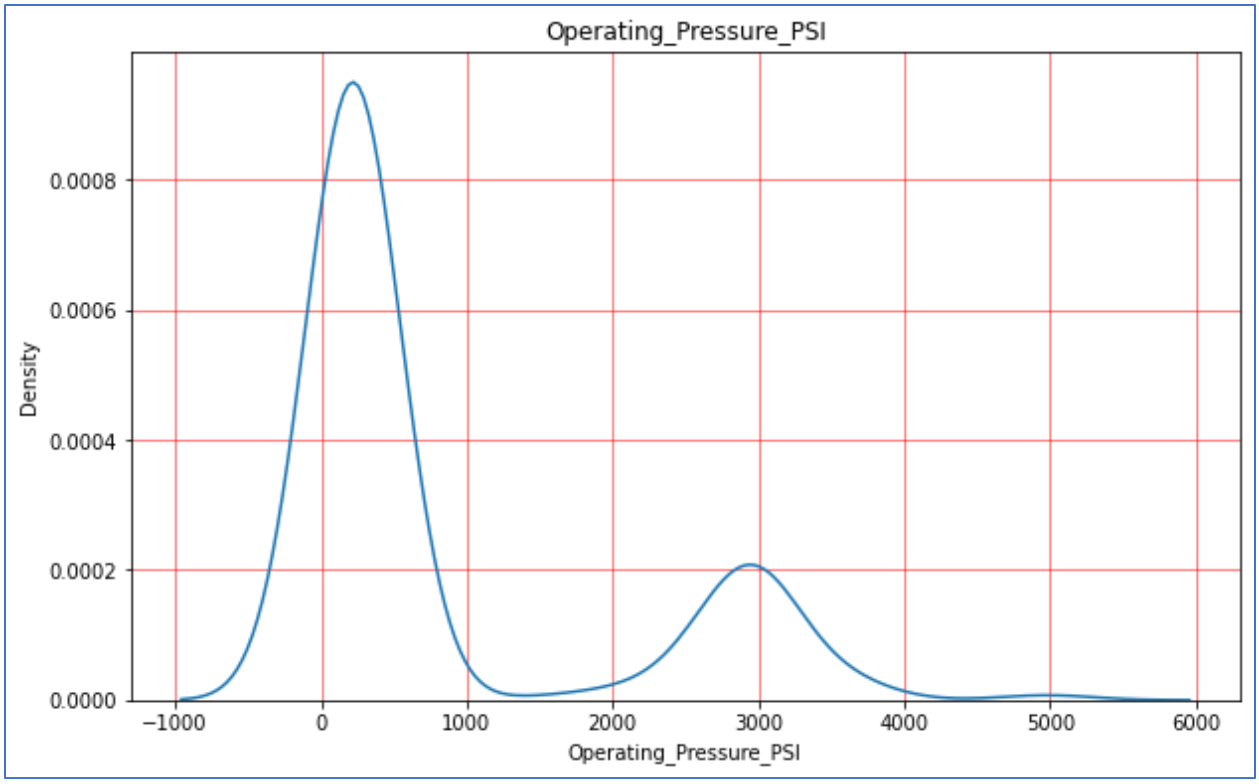

Figure 5: Distribution of Attr2_OpPr

Based on the distribution, the numerical features in the input data have been pre-processed:

1) costamount_usd, Rod_D, Bore_ID: Considering the distribution is close to normal, these columns will be normalized
2) Stroke & Operating_Pressure_PSI: Considering the data does not follow a specific distribution, split the data into quantiles and scale

Model building

The objective of the model is to create clusters based on the input data. As shown in figure 10, there are

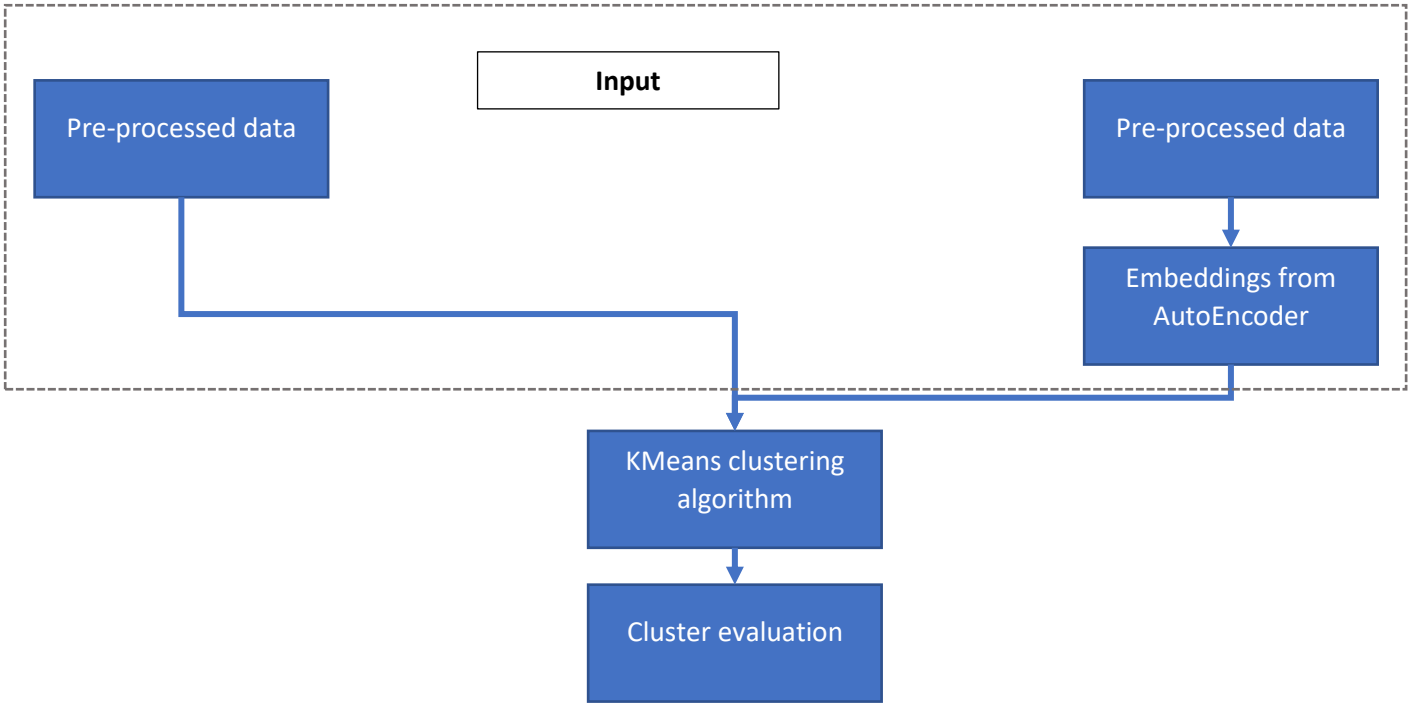

Figure 10: Clustering Model

2 different approaches have been utilized for generating the input data used for creating the clusters. The quality of the output clusters from the different scenarios have been evaluated to the select a particular approach.

Explanation of main algorithms

The two main algorithms used in this model Autoencoder and K-means. Autoencoder is used to generate embeddings of the input data. K-means is used to create clusters of the input data.

K-means algorithm[5]

K-means is an unsupervised clustering algorithm. K-means is an iterative algorithm. While the number of observations and dimensions of the input data are known, the user is required to pick the number of clusters. The cluster count is an important input to the K-means algorithm. The iterative steps are as follows:

1) Based on user input for cluster count n, n points from the input observations are randomly selected as cluster centroids
2) Each input observation is assigned to the nearest centroid
3) New cluster centroids are computed by taking the mean of all the points assigned to a particular cluster
4) Steps 2 & 3 continue until one of the following stop conditions is satisfied:

[5] https://databookuw.com/

a. Number of iterations has equaled the threshold
b. Cluster centroids are not changing versus the penultimate iteration

K-means is one of the simplest and effective clustering algorithms. Scikit-learn library provides out-of-the-box python implementations of the algorithm.

Autoencoder[5]

The primary objective of the Autoencoder is to learn a representation of the input data typically for the purpose of dimensionality reduction. The representations learned by the Autoencoder have practical applications such as clustering, recommendations etc. Autoencoder is implemented using neural networks. The Autoencoder architecture consists of an encoder and decoder. The encoder and decoder are both neural networks. The output of the encoder is the new representation of embedding of the input data. The output of the encoder acts as the input to the decoder. The decoder is expected to reconstruct the input data. Therefore, the training loss is the gap between the decoder and the original data. In this project, Autoencoder has been implemented using the tensorflow/keras library.

Hyperparameters

The most suitable configuration of the clustering model for this case study will be selected based on the results from 12 experiments. There are 2 main hyperparameters. The first one is the source of the input data. As shown in figure 10, the input data can be the plain pre-processed input data or the embeddings generated from the AutoEncoder. There are 3 different configurations considered for the AutoEncoder. The second experiment variable is the options for the cluster count. There are 3 different values considered for the cluster count. The list of experiments considered in this case study are shown in figure 11.

Challenges and improvements

During code development for the project, there were several practical challenges. Simple solutions and workarounds were utilized to overcome these challenges.

1) Autoencoder implementation: The initial implementation replicated the Autoencoder block for each hyperparameter combination. This approach unnecessarily expanded the code and made it difficult to track the results. Implementing the Autoencoder as a class and instantiating the class for each set of hyperparameters resulted in compact and manageable code
2) Correlation plots: Correlation heatmaps have been used extensively during EDA. When there are several numeric attributes, the correlation plot becomes hard to read and derive insights. In such instances, after the initial review of the correlation plot, selecting a subset of attributes to include in the correlation plot is helpful. This can be accomplished by selecting a minimum threshold for correlation value
3) Experiment tracking: There are 12 different experiments tracked in this project. The initial approach was to run each experiment individually and consolidate the results. Such an approach results in extra-long code and difficulty to review the results. An alternative approach is utilized in this project. The hyperparameters for each experiment is compiled in a dictionary. A specific function has been implemented to run all the experiments from experiment dictionary and consolidate the results. This simplified the review the experiment results and the overall code structure

Experiment number	Hyperparameter 1: Input	Hyperparameter 2: Cluster count
1	Pre-processed data	4
2	Pre-processed data	6
3	Pre-processed data	8
4	AutoEncoder (Embedding size = 4)	4
5	AutoEncoder (Embedding size = 4)	6
6	AutoEncoder (Embedding size = 4)	8
7	AutoEncoder (Embedding size = 6)	4
8	AutoEncoder (Embedding size = 6)	6
9	AutoEncoder (Embedding size = 6)	8
10	AutoEncoder (Embedding size = 8)	4
11	AutoEncoder (Embedding size = 8)	6
12	AutoEncoder (Embedding size = 8)	8

Figure 11: Experiments for clustering

Results & comparison table

The experiment results are shown in figure 12. Experiment_1.2 shows the best overall cluster quality. Experiment_1.2 belongs to the simplest group of scenarios and involves direct clustering of the pre-processed data.

Experiment number	Name	Training Loss	Silhouette Score	Calinski Harabasz Score	Davies Bouldin Score
0	Experiment_1.1	0.0000	0.3232	557.8044	1.0660
1	Experiment_1.2	0.0000	0.3598	544.8101	0.9070
2	Experiment_1.3	0.0000	0.3098	501.3407	1.0119
3	Experiment_2.1	0.0008	0.2694	351.6826	1.2175
4	Experiment_2.2	0.0012	0.2706	415.8404	1.1357
5	Experiment_2.3	0.0009	0.2471	362.8166	1.3458
6	Experiment_3.1	0.0002	0.3428	469.4927	0.9601
7	Experiment_3.2	0.0000	0.2732	393.0455	1.2373
8	Experiment_3.3	0.0001	0.2509	378.9276	1.4002
9	Experiment_4.1	0.0001	0.3016	410.2725	0.9816
10	Experiment_4.2	0.0001	0.2571	390.1845	1.2173
11	Experiment_4.3	0.0001	0.3202	463.9392	1.0491

Figure 12: Experiment results

Among the experiments utilizing embeddings, Experiment_3.1 shows the best overall cluster quality.

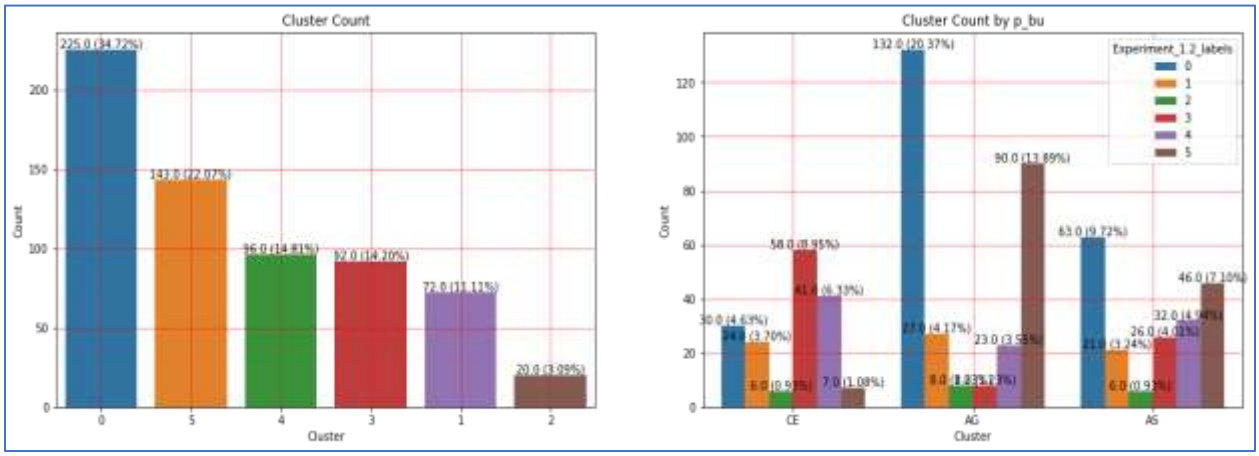

Figure 13: Cluster details by region and business unit

Out of the 6 clusters, top 2 clusters constitute 57% of the SKUs. At the BU level, there are 3 clusters with that have >5% SKUs across each BU.

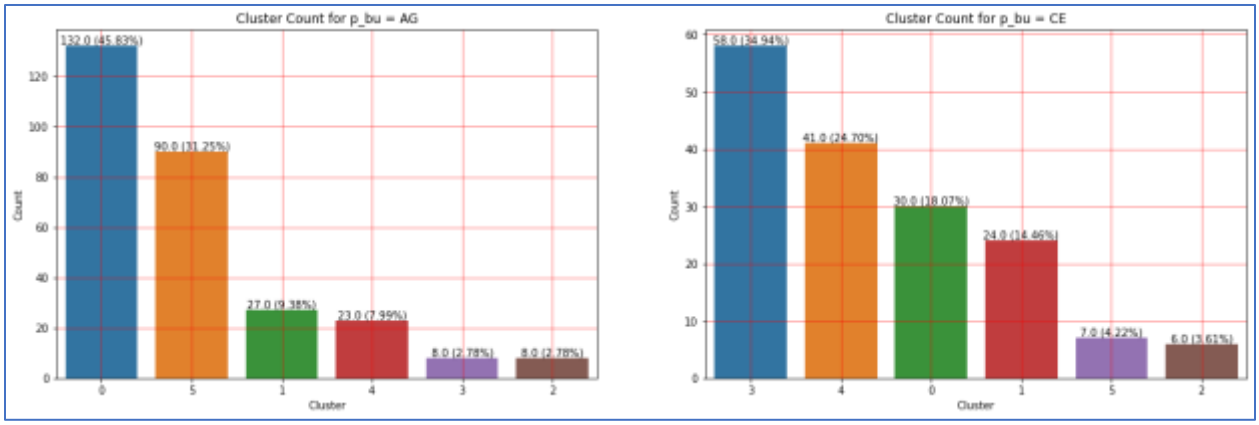

Figure 14: Cluster stratification by region & business unit

For BU = AG, there are 2 clusters that have > 15% of SKUs and the top 2 clusters contain 76% of the total SKUs.

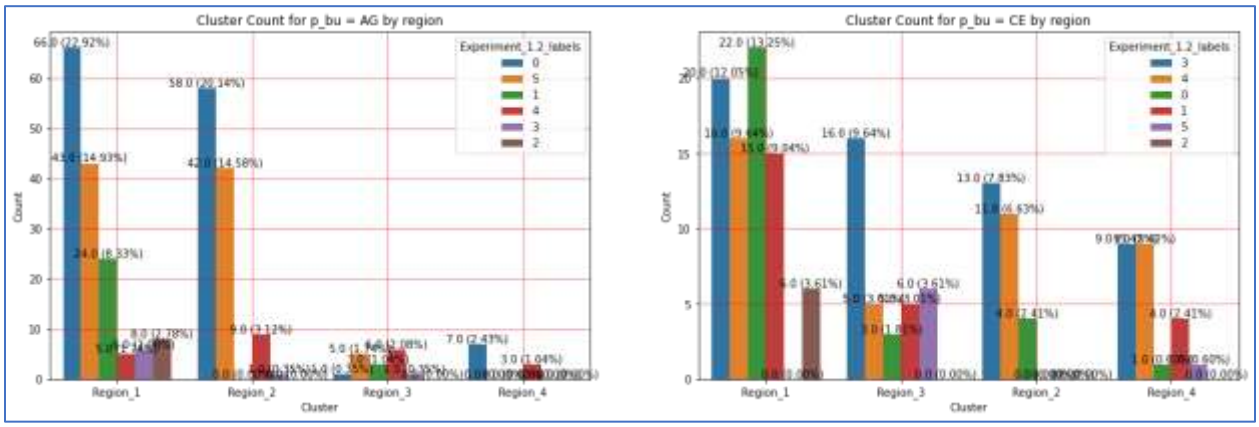

Figure 15: Cluster stratification by region for each business unit

For BU = AG, Region_1 and Region_2 have the same 2 top clusters making up ~35% of the SKUs. However, Region_1 has notable count of SKUs from a 3rd cluster [8% of SKUs]. For BU = CE, there are 3 clusters that have > 15% of SKUs and the top 3 clusters contain 75% of the total SKUs. For BU = CE, Region_1 has notable number of SKUs across 4 clusters, while the other regions have 1 or 2 main clusters

Conclusion

Experiment_1.2 has shown the best overall performance across the 3 metrics. It is ranked 1 on the Silhouette and Davies Boudlin scores. It is ranked 2 on the Calinski Harabasz score. Based on the results of Experiment_1.2 the following 6 clusters have been identified:

1) Cluster 0: Cluster with most SKUs (35%), used mainly in AG.
2) Cluster 1: Cluster with 11% SKUs, used in all BUs
3) Cluster 2: Cluster with the least SKUs (3%), used mainly in AG BU
4) Cluster 3: Cluster with 14% SKUs, top cluster in CE BU
5) Cluster 4: Cluster with 15% SKUs, used in all BUs, 2nd ranked cluster for CE

6) Cluster 5: Cluster with 22% SKUs, used mainly in AG BU

Acknowledgements

I am grateful to the following resources that were beneficial for this project:

Autoencoder Feature Extraction for Classification: https://machinelearningmastery.com/autoencoder-for-classification/

Intro to Autoencoders: https://www.tensorflow.org/tutorials/generative/autoencoder

Advance machine learning course on Clustering: https://developers.google.com/machine-learning/clustering

Suggested improvements

1) Gather additional data on SKUs and re-run the experiments: The purpose of this step would be to review if the cluster results can be improved using the embeddings from the AutoEncoder
2) Clustering algorithms: One of the suggestions would be to evaluate other clustering algorithms such as DBSCAN