Enterprise Resource Planning Blogs by SAP

Get insights and updates about cloud ERP and RISE with SAP, SAP S/4HANA and SAP S/4HANA Cloud, and more enterprise management capabilities with SAP blog posts.

Supplier clustering using Machine learning on Invoice dataset - Proof of concept

P

Puneet_Sikarwar

Product and Topic Expert

2021 Dec 29 12:34 PM

SAP Managed Tags: Machine Learning, SAP S/4HANA, MM Purchasing

Background:

Last week, I was giving a demo to customer on “Supplier segmentation & evaluation” in area of procurement & sourcing in SAP S/4HANA (on-premise). While explaining newly introduced feature purchasing categories via SAP FIORI app "Manage Purchasing Categories" , customer raised a very valid question.

Customer question – We have a supplier base of 1 million & some of them are dormant from months, most of them are involved in low business value transactions etc. We really do not know, how best we can cluster suppliers based on available invoice data. With limited IT team bandwidth we cannot perform classification of all 1 million suppliers.

This question motivated me to perform a POC(proof-of-concept) to cluster suppliers based on available Invoice data with the help of ML unsupervised algorithm. It does not make lot of sense to create purchasing categories & perform supplier evaluation

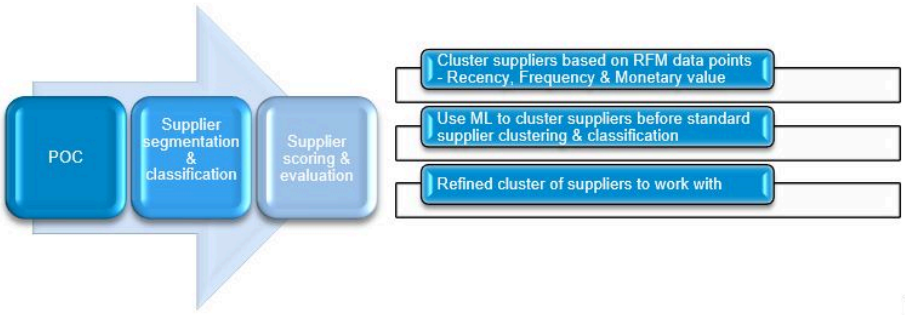
on all the suppliers. This POC is additional mechanism to filter non strategic suppliers before actual supplier classification & evaluation.

The idea is to show customer that using ML unsupervised learning algorithm, suppliers can be screened & supplier evaluation can be performed on smaller number of strategic suppliers to save time & effort of purchasing department.

12.8

To accomplish this POC, I have written Python code with ML KMeans algorithm to perform clustering of supplier before actual supplier classification & segmentation phase. Coding is done in Jupyter notebook & to understand this POC it is absolutely not necessary to understand python code. I have added code snippet for technical folks. Optional code snippet is added in each step under section “POC Execution”.

Objective:



The Aim of this POC is to screen suppliers based on RFM (Recency, Frequency, Monetary Value) parameters before performing actual supplier classification. It is first & basic level of refinement before creating purchasing categories in SAP S/4HANA(on premise) system. Below is more details of RFM methodology.

- R (Recency): Number of days since last invoice raised by supplier to check from how long the supplier is dormant.

Ex - How recent supplier have supplied the material - 50 Days

- F (Frequency): Number of transactions per supplier in given time frame

Ex - How Frequently procurement is done with supplier - 100 invoices raised in given timeframe

- M (Monetary Value) - How much procurement done with supplier in $

Ex - Total 100$ worth of invoices raised by supplier done in given timeframe

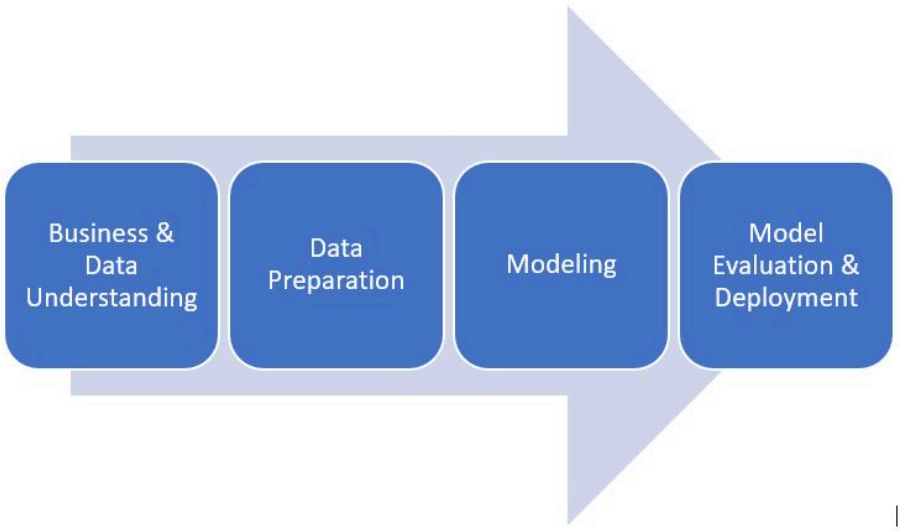
Business Value:

- Reduce workload of supplier screening & evaluation team as we get refined list of strategic suppliers after the exercise.
- Identify strategically important suppliers which fall under high RFM cluster. Supplier with recent, high value monetary transactions.
- Improves accuracy of supplier scoring & evaluation phase.

POC Execution:

This problem is a clear candidate of Unsupervised machine learning use case as Output/Label is not predefined & we are expected to Cluster suppliers using ML Algorithm.

Implementation of data science POC from scratch tend to become too technical, I am not denying that but there is equal involvement of functional counterpart in each stage. At each stage I will share important points which a functional expert can drive & discuss with customer.



-

Business & data understanding

Business understanding is shared in section **Background & objective** of POC.

List of 500k Invoice transactions taken from Jan 2019 to Dec 2020, sample Excel data:

	A	B	C	D	E	F	G	H
	InvoiceNo	Material Code	Description	Quantity	InvoiceDate	UnitPrice	Invoicing Party	Country
	10000001	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	1/12/2019 8:26	2.55	17850	United Kingdom
	10000001	71053	WHITE METAL LANTERN	6	1/12/2019 8:26	3.39	17850	United Kingdom
	10000002	84406B	CREAM CUPID HEARTS COAT HANGER	8	1/12/2019 8:26	2.75	17850	United Kingdom
	10000002	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	1/12/2019 8:26	3.39	17850	United Kingdom
	10000003	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	1/12/2019 8:26	3.39	17850	United Kingdom

Columns :

- InvoiceNo. - Unique invoice number (Under 1 Invoice number multiple material code are invoiced, Hence more than 1 row exist per invoice in dataset)

- Material Code – Material Invoiced

- Invoicing party - Supplier
- Quantity – Invoice quantity of material code invoices at UnitPrice.
- InvoiceDate – invoicing date with timestamp

Optional* - Read excel, check data types & important statistical parameters

In [2]:
```
# read the dataset
retail_df = pd.read_csv("Online+Retail.csv", sep=",", encoding="ISO-8859-1", header=0)
retail_df.head()
```

Out[2]:

	InvoiceNo	Material Code	Description	Quantity	InvoiceDate	UnitPrice	Invoicing Party	Country
0	100536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	1/12/2019 8:26	2.55	17850.0	United Kingdom
1	100536365	71053	WHITE METAL LANTERN	6	1/12/2019 8:26	3.39	17850.0	United Kingdom
2	100536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	1/12/2019 8:26	2.75	17850.0	United Kingdom
3	100536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	1/12/2019 8:26	3.39	17850.0	United Kingdom
4	100536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	1/12/2019 8:26	3.39	17850.0	United Kingdom

In [3]:
```
# basics of the df
retail_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541913 entries, 0 to 541912
Data columns (total 8 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   InvoiceNo        541913 non-null  object
 1   Material Code    541911 non-null  object
 2   Description      540459 non-null  object
 3   Quantity         541913 non-null  int64
 4   InvoiceDate      541913 non-null  object
 5   UnitPrice        541913 non-null  float64
 6   Invoicing Party  406831 non-null  float64
 7   Country          541913 non-null  object
dtypes: float64(2), int64(1), object(5)
memory usage: 33.1+ MB
```

Data Preparation

Data preparation is very important step before model building. Following are important questions answered in this step.

- What to do with missing values in the dataset(Excel in this case) - Ex - Delete or replace missing data with mean/median/mode

Most of the times data may not be complete & is plagued with missing values. Hence functional consultant are expected to discuss missing value issue before model building. It is required to treat missing values in dataset by either deletion of columns or rows with missing values Or by replacing them with statistical mean/mode/median. There are lot of ways of dealing missing values however in this POC we have dropped rows with missing values as there was no scarcity of data.

Functional consultant is expected to discuss treatment of missing values with customer.

Optional* - Code snippet to highlight % of missing values & dropping rows with missing values.

```
In [6]: # missing values in %
        round(100*(retail_df.isnull().sum())/len(retail_df), 2)

Out[6]: InvoiceNo          0.00
        Material Code      0.00
        Description        0.27
        Quantity           0.00
        InvoiceDate        0.00
        UnitPrice          0.00
        Invoicing Party   24.93
        Country            0.00
        dtype: float64

In [ ]:

In [7]: # drop all rows having missing values
        retail_df = retail_df.dropna()
        retail_df.shape

Out[7]: (406829, 8)
```

- Drive important features/columns out of existing dataset columns. Ex - Recency, Frequency, Monetary

This is very important step as part of data preparation as it involves creation of new features/columns to increase insights in given dataset. To perform supplier clustering on RFM (Recency, Frequency, Monetary) parameters it is required to create three additional columns (amount, recency & frequency) to depict recency,frequency & monetary value by grouping existing data.

Monetary Value: Total invoice transaction amount per supplier – Group

by supplier on Amount field (calculated by multiplying Quantity * Unit Price)

Optional*

Aggregating based on RFM

- R (Recency): Number of days since last purchase
- F (Frequency): Number of tracsactions
- M (Monetary): Total amount of transactions (revenue contributed)

In [11]:
```
# monetary
grouped_df = retail_df.groupby('Invoicing Party')['amount'].sum()
grouped_df = grouped_df.reset_index()
grouped_df.head()
```

Out[11]:

	Invoicing Party	amount
0	12346.0	0.00
1	12347.0	4310.00
2	12348.0	1797.24
3	12349.0	1757.55
4	12350.0	334.40

Frequency: Frequency of invoice transaction per supplier – Group by supplier & count transaction per supplier.

Optional*

In [13]:
```
# merge the two dfs
grouped_df = pd.merge(grouped_df, frequency, on='Invoicing Party', how='inner')
grouped_df.head()
```

Out[13]:

	Invoicing Party	amount	frequency
0	12346.0	0.00	2
1	12347.0	4310.00	182
2	12348.0	1797.24	31
3	12349.0	1757.55	73
4	12350.0	334.40	17

Recency: Last Invoice transaction done by supplier from today in Days.

Optional*

In [19]:
```
# recency
last_purchase = retail_df.groupby('Invoicing Party')['diff'].min()
last_purchase = last_purchase.reset_index()
last_purchase.head()
```

Out[19]:

	Invoicing Party	diff
0	12346.0	326 days 02:33:00
1	12347.0	1 days 20:58:00
2	12348.0	74 days 23:37:00
3	12349.0	18 days 02:59:00
4	12350.0	310 days 20:49:00

Final dataset prepared based on RFM analysis.

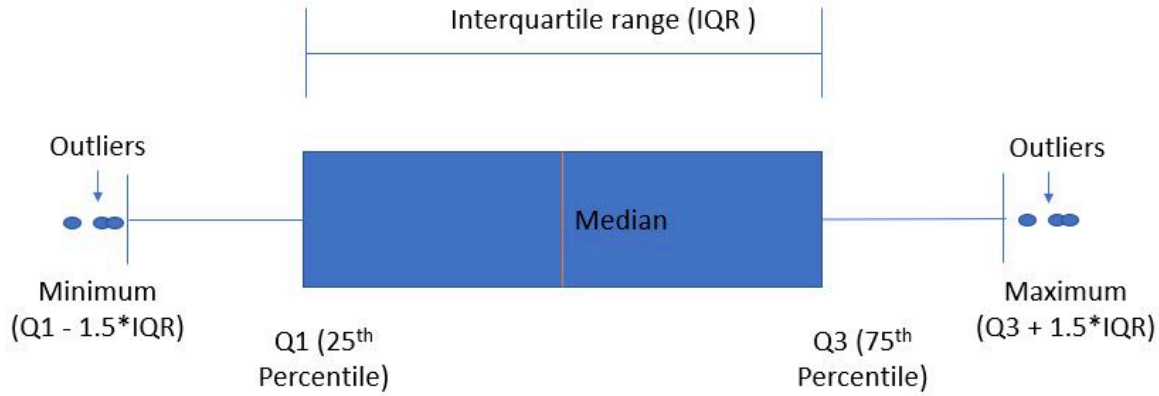
In [20]:
```
# merge
grouped_df = pd.merge(grouped_df, last_purchase, on='Invoicing Party', how='inner')
grouped_df.columns = ['Invoicing Party', 'amount', 'frequency', 'recency']
grouped_df.head()
```

Out[20]:

	Invoicing Party	amount	frequency	recency
0	12346.0	0.00	2	326 days 02:33:00
1	12347.0	4310.00	182	1 days 20:58:00
2	12348.0	1797.24	31	74 days 23:37:00
3	12349.0	1757.55	73	18 days 02:59:00
4	12350.0	334.40	17	310 days 20:49:00

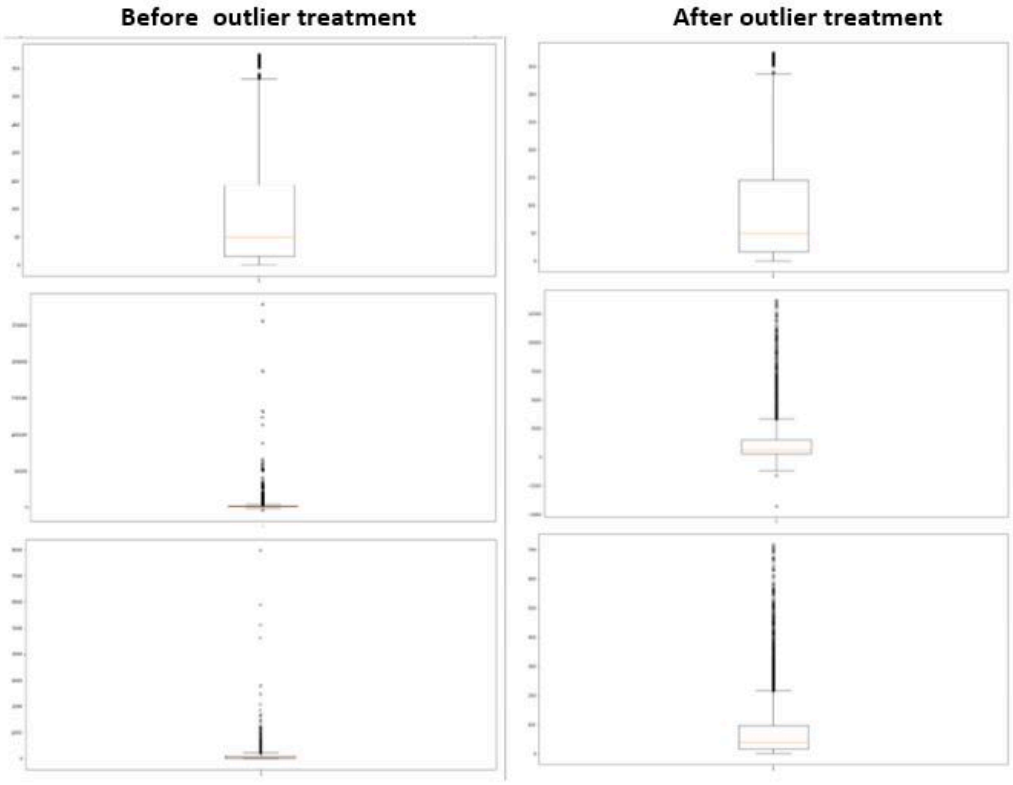
- How outliers are treated ?

Outliers are data points which holds values less & more than lower band & higher band of boxplot. Below is depiction of outliers in statistical terms



Outliers highly influence clustering of suppliers, Hence treatment of outliers are important before model building & driving new columns.

In this POC since ample amount of dataset was available I just deleted outliers present in column Recency, Frequency & Monetary value.



- Share the Insights with customer after exploratory data Analysis.

Ex - Highest number of Tx observed in UK Or Top 10 suppliers with business value

Modeling, evaluation & deployment

Determination is optimum number of clusters precedes model building & evaluation.

There are methods like Silhouette Analysis & elbow curve to determine optimum number of clusters. However, it is always wise to take first feedback from customer on number of clusters. Based on business discussion primarily on customer workload bandwidth, hence customer advised to have 3 clusters.

Multiple unsupervised learning algorithms like KMeans, Random forest, Decision tree can be used. Before finalizing algorithms for clustering it is important to understand pros & cons of each of them. Explaining pros & cons of each of them is a huge topic in

itself & hence not explained here. For this POC we used KMeans clustering algorithm. Below is the summary

- No. of cluster – 3, tagged as (0,1,2)
- Model employed – Kmeans
- Model learning Iterations – 50

Optional*

```
In [27]: # k-means with some arbitrary k
         kmeans = KMeans(n_clusters=3, max_iter=50)
         kmeans.fit(rfm_df_scaled)

Out[27]: KMeans(max_iter=50, n_clusters=3)
```

```
In [34]: # assign the label
         grouped_df['cluster_id'] = kmeans.labels_
         grouped_df.head()
```

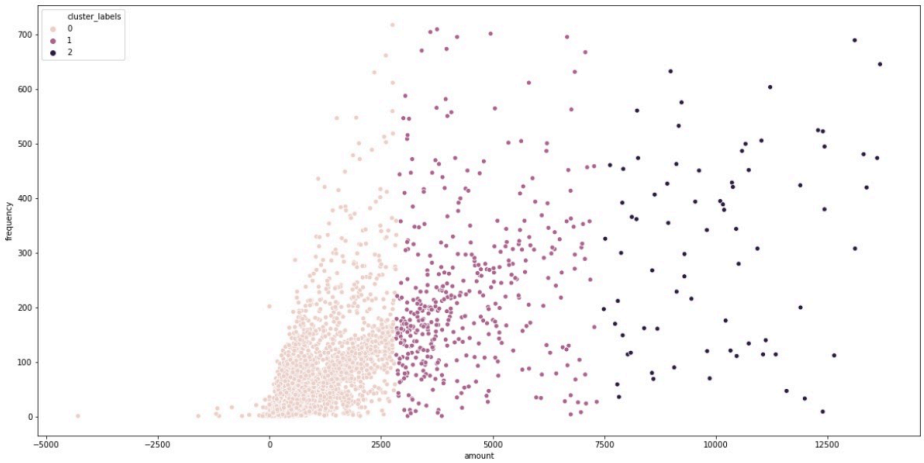
Out[34]:

	Invoicing Party	amount	frequency	recency	cluster_id
0	12346.0	0.00	2	326	1
1	12347.0	4310.00	182	1	0
2	12348.0	1797.24	31	74	0
3	12349.0	1757.55	73	18	0
4	12350.0	334.40	17	310	1

Graphical display of Clusters

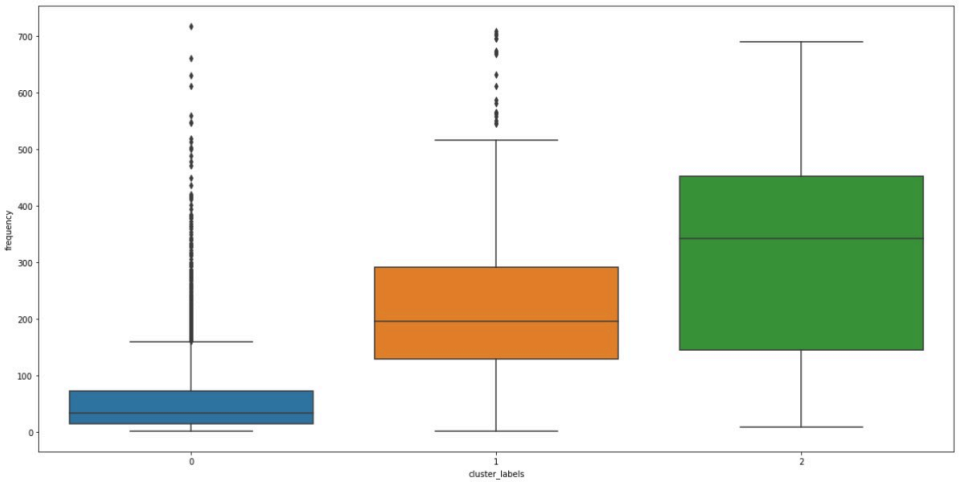
Frequency – Scatter plot of clusters (0,1,2)

In [44]:
```
plt.figure(figsize=(20,10))
sns.scatterplot(x='amount', y='frequency', hue ='cluster_labels', data=grouped_df)
plt.show()
```



In [42]:
```
# plots
plt.figure(figsize=(20,10))
sns.boxplot(x='cluster_labels', y='frequency', data=grouped_df)
```

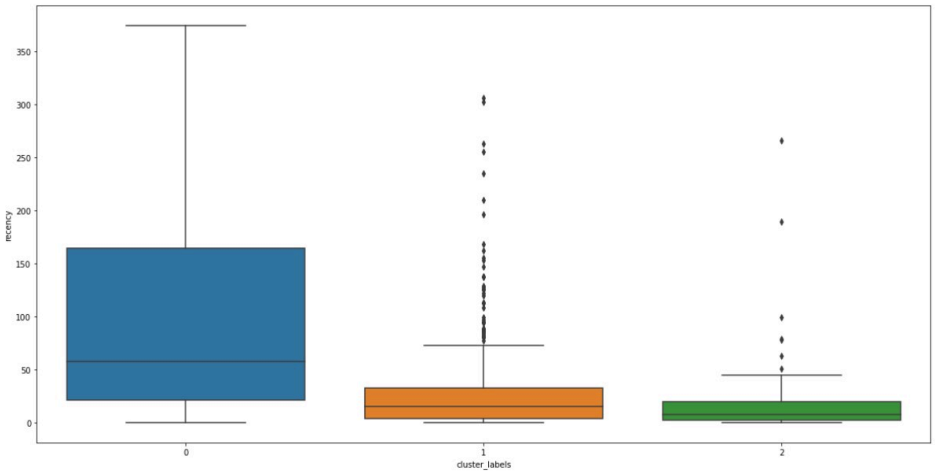
Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x1990ed83700>



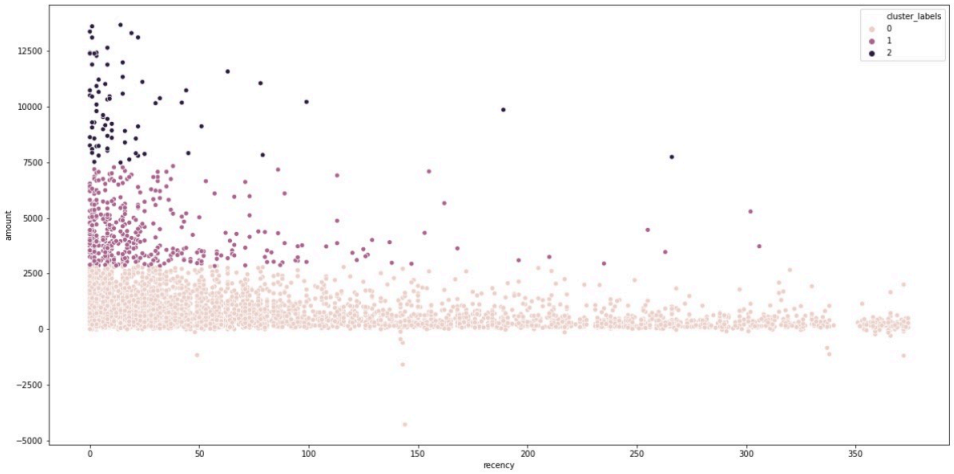
Recency – Scatter plot of clusters (0,1,2)

In [41]:
```
# plots
plt.figure(figsize=(20,10))
sns.boxplot(x='cluster_labels', y='recency', data=grouped_df)
```

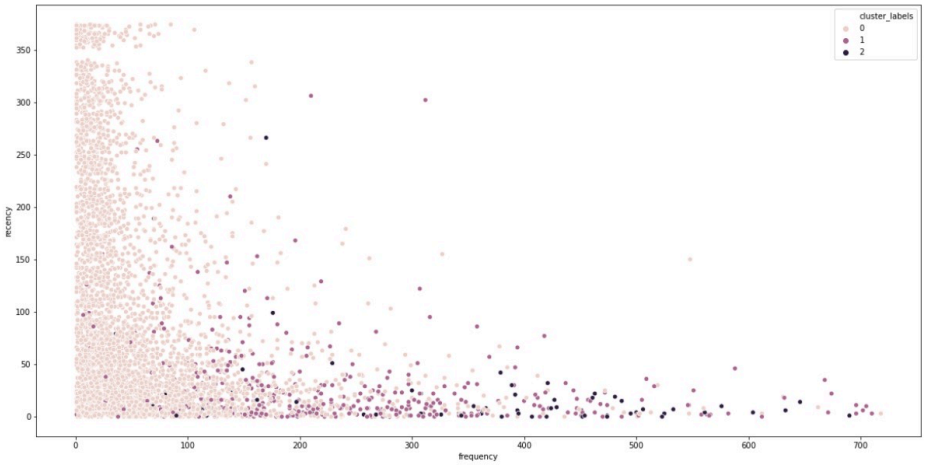
Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x1991541e880>



```
In [45]: plt.figure(figsize=(20,10))
         sns.scatterplot(y='amount', x='recency', hue ='cluster_labels', data=grouped_df)
         plt.show()
```

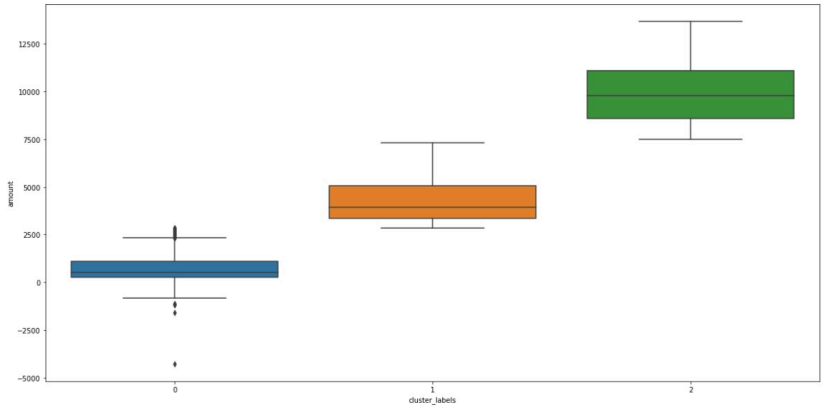



```
In [46]: plt.figure(figsize=(20,10))
         sns.scatterplot(x='frequency', y='recency', hue ='cluster_labels', data=grouped_df)
         plt.show()
```

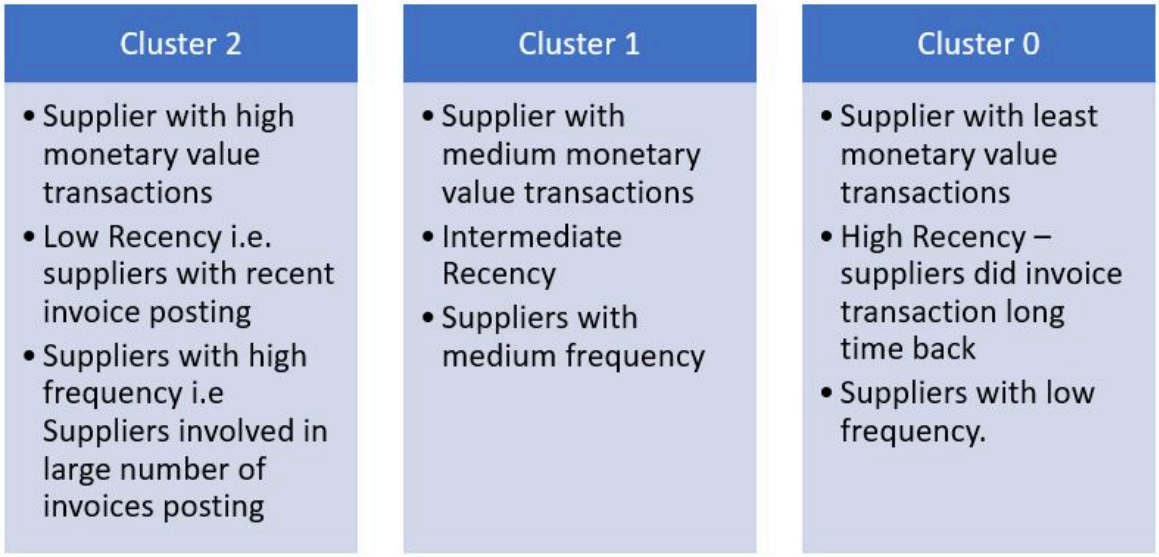


Amount/Monetary Value– Scatter plot of clusters (0,1,2)

```
In [43]: # plots
         plt.figure(figsize=(20,10))
         sns.boxplot(x='cluster_labels', y='amount', data=grouped_df)
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x199157b19d0>
```



Inference & Conclusion:



Above inference highlights suppliers tagged under Cluster 2 are strategically important as they exhibit frequent high-volume transaction. Hence suppliers tagged under cluster 2 should be taken in follow on process like supplier classification & evaluation.

Important consideration

- Other unsupervised learning algorithms like RF, DT etc. can be also employed, They may produce similar or even better result however I have not tried them in this POC
- I had taken invoice data for clustering & have not included Credit or Debit Memo for simplicity. However same can included.
- Here customer has guided us on number of clusters considering IT team bandwidth & other business aspects, however 3 or more clusters can be created.
- Number of clusters can also be determined based on Elbow curve or silhouette score.
- Dataset displayed in screen shots are dummy dataset.
- Some of the decisions taken in this proof of concept like deletion of missing values, selection of model etc. varies a lot with dataset, Hence should not be followed as it is.

Important links

- RFM Segmentation of customer in SAP CRM https://help.sap.com/viewer/fa2412b6dfa4460c9b840168d22782ff/7.57.30/en-US/09f0895360b93d58e10000000...

Labels:

Life at SAP

1 Comment

M

Mohan_Avasaram
Explorer

2021 Dec 29 6:25 PM

1 Kudo

Excellent work! Thanks for sharing.

You must be a registered user to add a comment. If you've already registered, sign in. Otherwise, register and sign in.

Comment

Labels In This Area

API and Integration 1 | Artificial Intelligence (AI) 1 | Business Trends 361 | Business Trends 97 | Cards 2 | Chart View 2 | Customer COE Basics and Fundamentals 1 | Date Offset 2 | Digital Transformation with Cloud ERP (DT) 1 | Event Information 451 | Event Information 77 | Expert Insights 109 | Expert Insights 529 | General 2 | General Overview 1 | Governance and Organization 1 | Great Britain 1 | Introduction 1 | Life at SAP 408 | Life at SAP 11 | List View 2 | Product Updates 4,598 | Product Updates 684 | Roadmap and Strategy 1 | Technology Updates 1,486 | Technology Updates 273 | User Defined Queries (UDQ) 1

Related Content

Text Embedding Service in SAP HANA Cloud Predictive Analysis Library (PAL)
in Enterprise Resource Planning Blogs by SAP 3 weeks ago

GROW with SAP S/4HANA Cloud Public Edition, premium - A Practical Guide
in Enterprise Resource Planning Blogs by SAP 2024 Dec 09

2024 SAP BTP Year in Review: Release Highlights for SAP S/4HANA users
in Enterprise Resource Planning Blogs by SAP 2024 Nov 18

Artificial Intelligence and other Innovations in SAP S/4HANA Cloud Private Edition 2023 FPS02
in Enterprise Resource Planning Blogs by SAP 2024 Nov 04

Understanding RISE with SAP: Simplifying the Confusion
in Enterprise Resource Planning Blogs by Members 2024 Oct 18

Popular Blog Posts

Useful documents on SCN

N **Nancy**
Product and Topic Expert

149013 123 225

Evolution of ABAP

karl_kessler
Product and Topic Expert

41288 45 214

Analytics in S/4HANA - real shape of embedded analytics and beyond embedded analytics

Masaaki
Product and Topic Expert

118119 32 190

Top Kudoed Authors

K	KripaKrishnadas	5
B	BethBerr	5
	Marco_Valencia	5
Y	LynetteSun	5
	Masaaki	4
E	EPantino	4

PrasanthPadmanabhan 4

Chr_Vogler 3

J **jmarquezmr** 3

J **Janet_Salmon** 3

View all