In [ ]:

Report

Sections required in your report:

- Main objective of the analysis that also specifies whether your model will be focused on clustering or dimensionality reduction and the benefits that your analysis brings to the business or stakeholders of this data.
- Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.
- Brief summary of data exploration and actions taken for data cleaning orfeature engineering.
- Summary of training at least three variations of the unsupervised model you selected. For example, you can use different clustering techniques or different hyperparameters.
- A paragraph explaining which of your Unsupervised Learning models you recommend as a final model that best fits your needs in terms.
- Summary Key Findings and Insights, which walks your reader through the main findings of your modeling exercise.
- Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model or adding specific data features to achieve a better model.

1) Main objective of the analysis that also specifies whether your model will be focused on clustering or dimensionality reduction and the benefits that your analysis brings to the business or stakeholders of this data.

In case of clustering, which is a branch of unsupervised learning, the main goal is to derive unseen structures from the data.
Hence to above, the goal of my analysis was to provide the best possible interpretation of what kind of clients we are dealing with.
If we know little about our clients or if we are looking after some less obvious subgroups of our clients then clustering can prove useful in deriving proper understanding of clients characteristics.
Above may help in planning more effective sales campaigns due to smarter targeting.
For the sake of visualization I will use PCA to get three components for 3-D scatter plot.

2) Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.

The CC GENERAL data set used here in the analysis comes from Kaggle:
https://www.kaggle.com/arjunbhasin2013/ccdata
The Dataset describes behavior of 8950 active credit card holders during last 6 months.
Each row shows eighteen behavioral variables of single customer.

In [109…

```
credit_cards.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Index: 8950 entries, C10001 to C19190
Data columns (total 18 columns):
 #   Column                            Non-Null Count  Dtype  
---  ------                            --------------  -----  
 0   BALANCE                           8950 non-null   float64
 1   BALANCE_FREQUENCY                 8950 non-null   float64
 2   PURCHASES                         8950 non-null   float64
 3   ONEOFF_PURCHASES                  8950 non-null   float64
 4   INSTALLMENTS_PURCHASES            8950 non-null   float64
 5   CASH_ADVANCE                      8950 non-null   float64
 6   PURCHASES_FREQUENCY               8950 non-null   float64
 7   ONEOFF_PURCHASES_FREQUENCY        8950 non-null   float64
 8   PURCHASES_INSTALLMENTS_FREQUENCY  8950 non-null   float64
 9   CASH_ADVANCE_FREQUENCY            8950 non-null   float64
 10  CASH_ADVANCE_TRX                  8950 non-null   float64
 11  PURCHASES_TRX                     8950 non-null   float64
 12  CREDIT_LIMIT                      8950 non-null   float64
 13  PAYMENTS                          8950 non-null   float64
 14  MINIMUM_PAYMENTS                  8950 non-null   float64
 15  PRC_FULL_PAYMENT                  8950 non-null   float64
 16  TENURE                            8950 non-null   float64
 17  cluster                           8950 non-null   int32  
dtypes: float64(17), int32(1)
memory usage: 1.6+ MB
```

More details on GitHub

3) Brief summary of data exploration and actions taken for data cleaning or feature engineering.

The EDA process involved:

- changing data types of variables from integers to floats
- dealing with NULL values:
  - There were only two variables with missing values:
    - "MINIMUM_PAYMENTS" had 313 missings values
    - "CREDIT_LIMIT" had one missing value
  - Both variables were filled with each variable's median
- checking and assessing strength of correlation between variable
  - eventually fives variables were kept to retrain clustering algorithms - variables with medium and weak correlations between each other were left:
    - "BALANCE"
    - "PURCHASES"
    - "ONEOFF_PURCHASES_FREQUENCY"
    - "CASH_ADVANCE_FREQUENCY"
    - "CREDIT_LIMIT"
- variables were normalized with a use of numpy's log1p()
- varaibles were standarized with a use of sklearn's MinMaxScaler()

More details on GitHub

NULL values:

In [110…

```
credit_cards = pd.read_csv('CC GENERAL.csv',index_col=["CUST_ID"])
```

```
credit_cards[credit_cards.select_dtypes(include=['int64', "uint8"]).columns] = credit_c
credit_cards.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 8950 entries, C10001 to C19190
Data columns (total 17 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   BALANCE                           8950 non-null   float64
 1   BALANCE_FREQUENCY                 8950 non-null   float64
 2   PURCHASES                         8950 non-null   float64
 3   ONEOFF_PURCHASES                  8950 non-null   float64
 4   INSTALLMENTS_PURCHASES            8950 non-null   float64
 5   CASH_ADVANCE                      8950 non-null   float64
 6   PURCHASES_FREQUENCY               8950 non-null   float64
 7   ONEOFF_PURCHASES_FREQUENCY        8950 non-null   float64
 8   PURCHASES_INSTALLMENTS_FREQUENCY  8950 non-null   float64
 9   CASH_ADVANCE_FREQUENCY            8950 non-null   float64
 10  CASH_ADVANCE_TRX                  8950 non-null   float64
 11  PURCHASES_TRX                     8950 non-null   float64
 12  CREDIT_LIMIT                      8949 non-null   float64
 13  PAYMENTS                          8950 non-null   float64
 14  MINIMUM_PAYMENTS                  8637 non-null   float64
 15  PRC_FULL_PAYMENT                  8950 non-null   float64
 16  TENURE                            8950 non-null   float64
dtypes: float64(17)
memory usage: 1.2+ MB
```

In [111…

```
missing_data_count = pd.DataFrame(credit_cards.isnull().sum().sort_values(ascending=Fal
missing_data_count.T
```

Out[111…

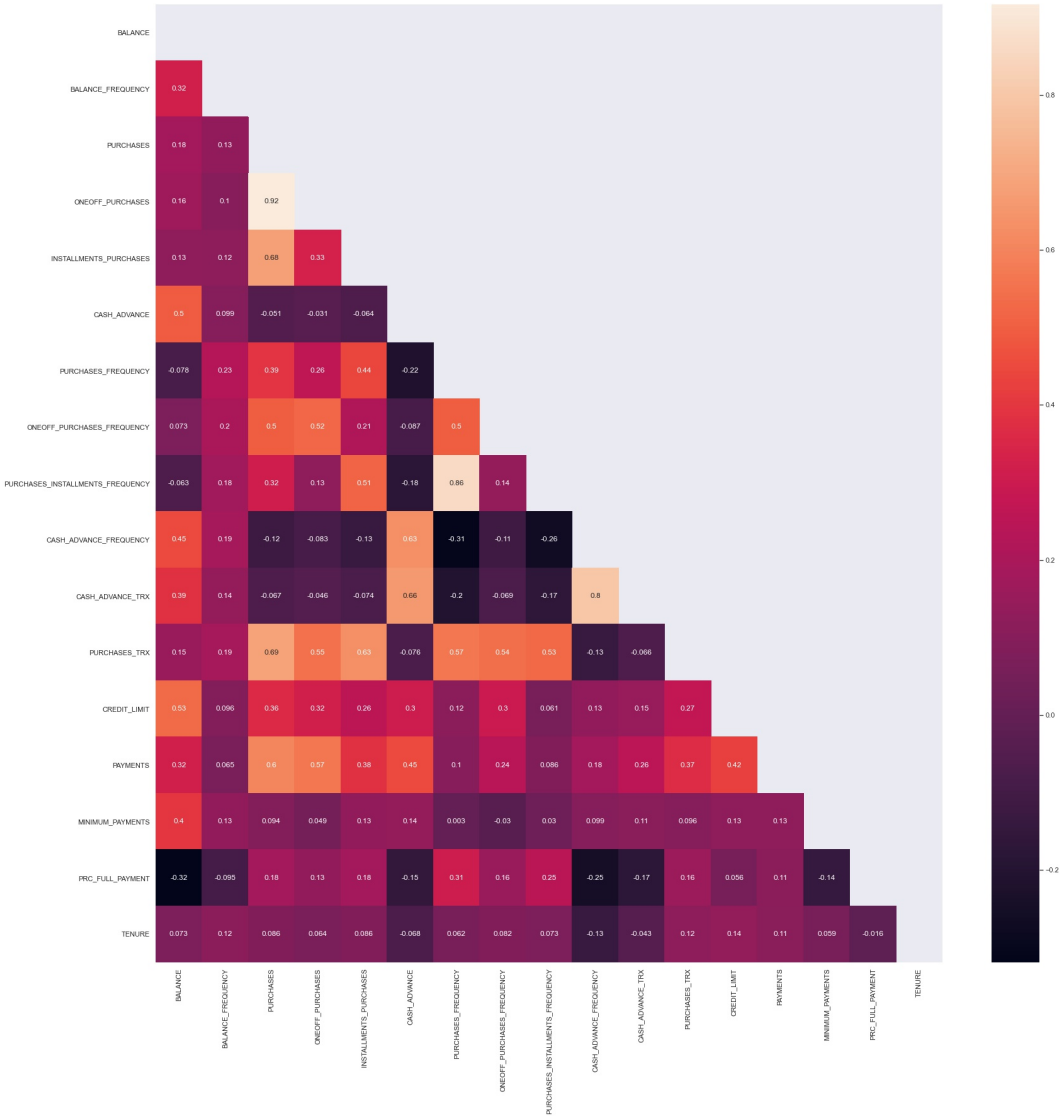
	MINIMUM_PAYMENTS	CREDIT_LIMIT	TENURE	PURCHASES_FREQUENCY	BALANCE_FREQUENCY	PUR
0	313	1	0	0	0	

Initial correlation matrix:

In [142…

```
Image(filename='ini_corr_matrix.jpg')
```

Out[142…

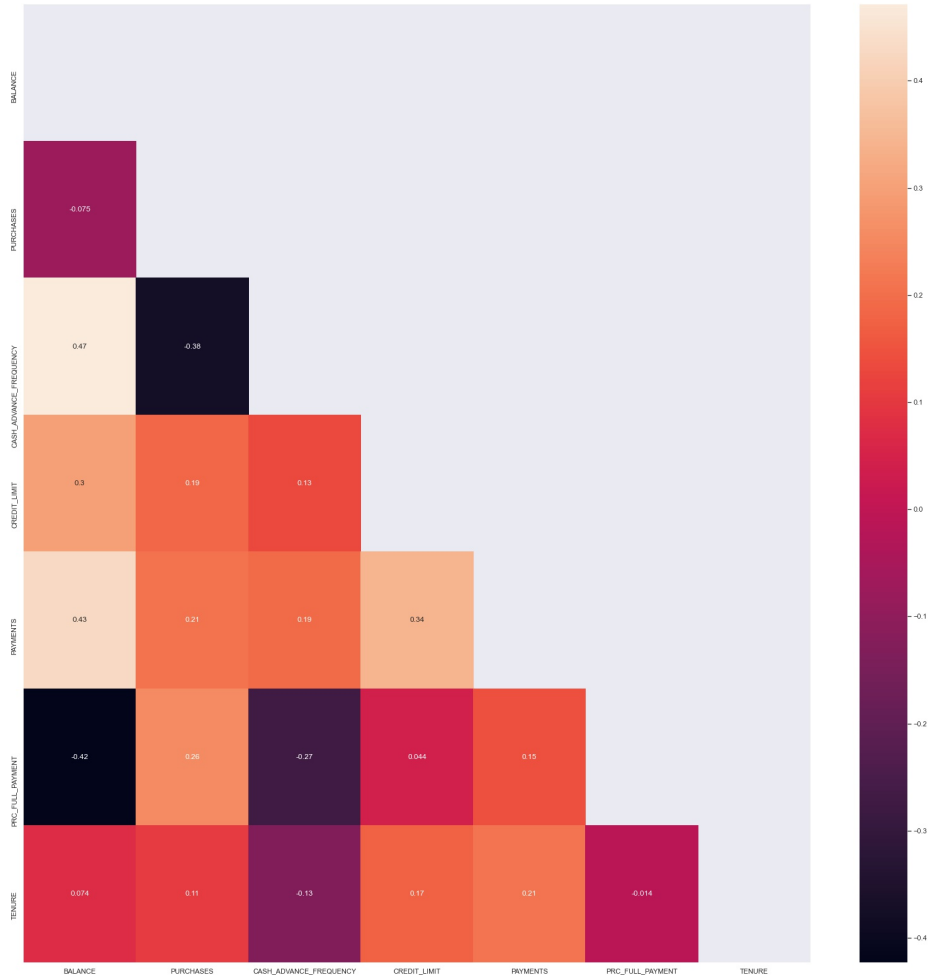


Reduced correlation matrix:

In [143…

```
Image(filename='corr_matrix_reduced_normalized_standarized.jpg')
```

Out[143…



4) Summary of training at least three variations of the unsupervised model you selected. For example, you can use different clustering techniques or different hyperparameters.

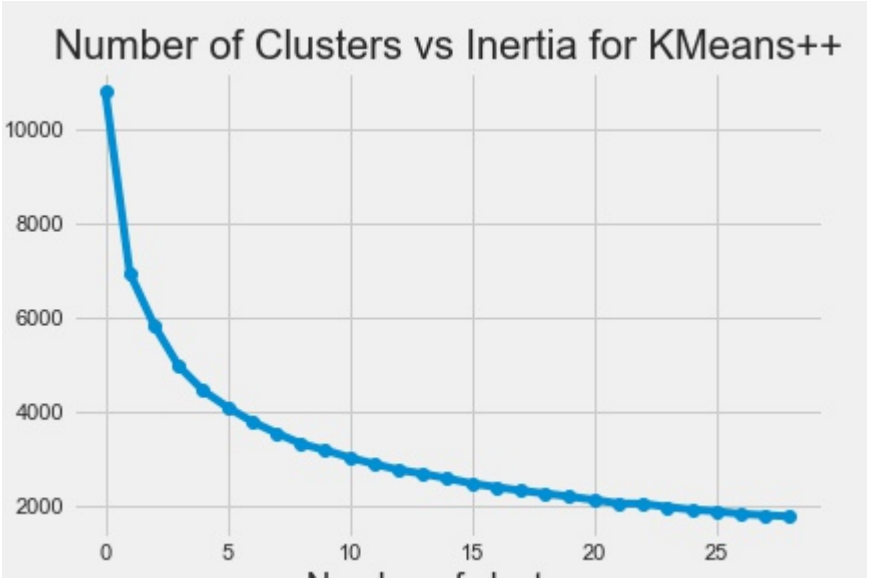
Inertia:

As a first step Inertia was calulated and visualised:

In [114...

```
Image(filename="inertia.jpg")
```

Out[114...



From Elbow rule we can tell that even 4-6 clusters might have had been hidden in our data set.
However, initial findings taken from distribution plots did not show any significant differences between distributions for cases when space was divided into more than three clusters.
The shapes of distributions were overlapping each other when they should show diffrent characteristics of skewness and kurtosis between clusters.
Due to above I decided first to fit the data set into two clusters, then three clustars, and more.
The goal was to decide by trial and error what is reasonably the largest number of clusters which still allows for visual discrimination of variables.

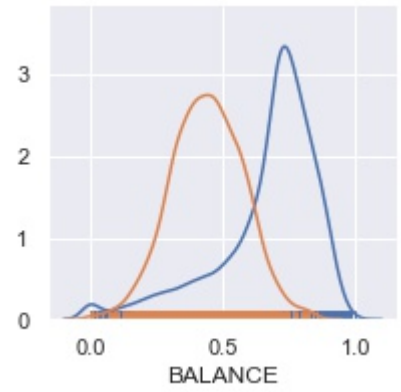
K-Means:

Few examples of distributions derived from k-means clustering trials:

Training two clusters:

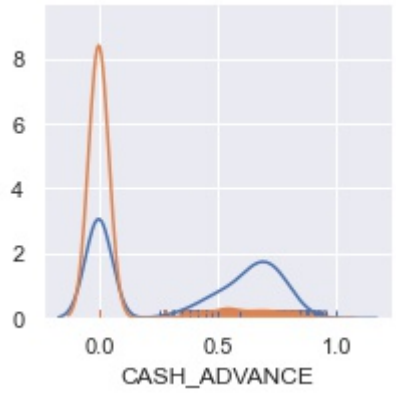
In [115…
```
Image(filename='two_clusters_distributions_of_BALANCE.jpg')
```

Out[115…



In [116…
```
Image(filename='two_clusters_distributions_of_CASH_ADVANCE.jpg')
```

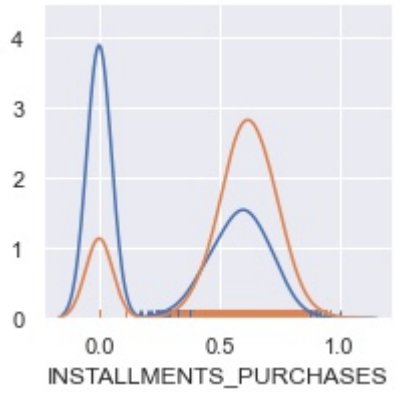
Out[116…



In [117…

```
Image(filename='two_clusters_distributions_of_INSTALLMENTS_PURCHASES.jpg')
```

Out[117…

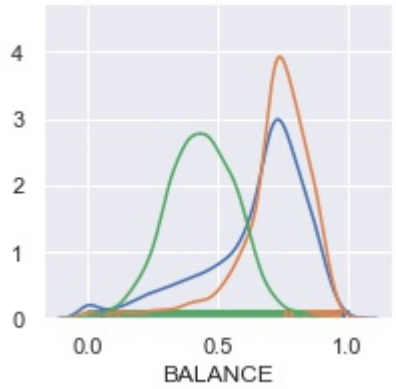


Training three clusters:

In [118…

```
Image(filename='three_clusters_distributions_of_BALANCE.jpg')
```

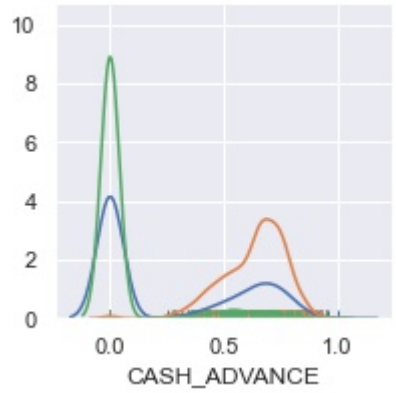
Out[118…



In [119…

```
Image(filename='three_clusters_distributions_of_CASH_ADVANCE.jpg')
```

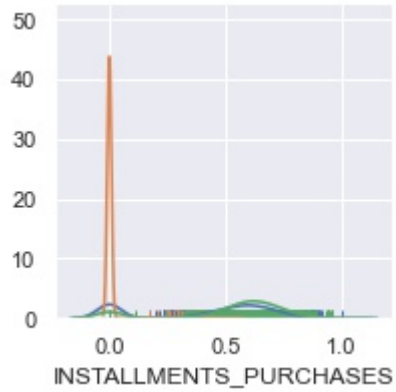
Out[119…



In [120…

```
Image(filename='three_clusters_distributions_of_INSTALLMENTS_PURCHASES.jpg')
```

Out[120…

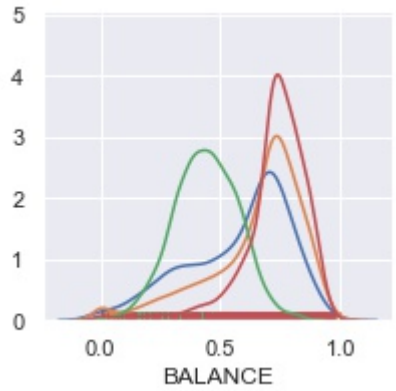


Training four clusters:

In [121…

```
Image(filename='four_clusters_distributions_of_BALANCE.jpg')
```

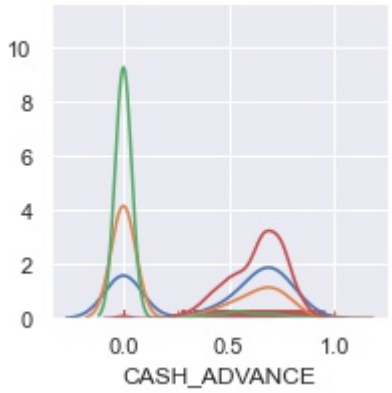
Out[121…



In [122…

```
Image(filename='four_clusters_distributions_of_CASH_ADVANCE.jpg')
```

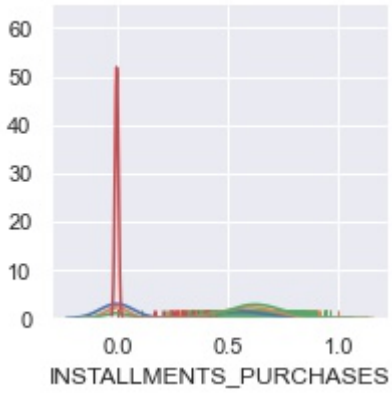
Out[122…



In [123…

```
Image(filename='four_clusters_distributions_of_INSTALLMENTS_PURCHASES.jpg')
```

Out[123…

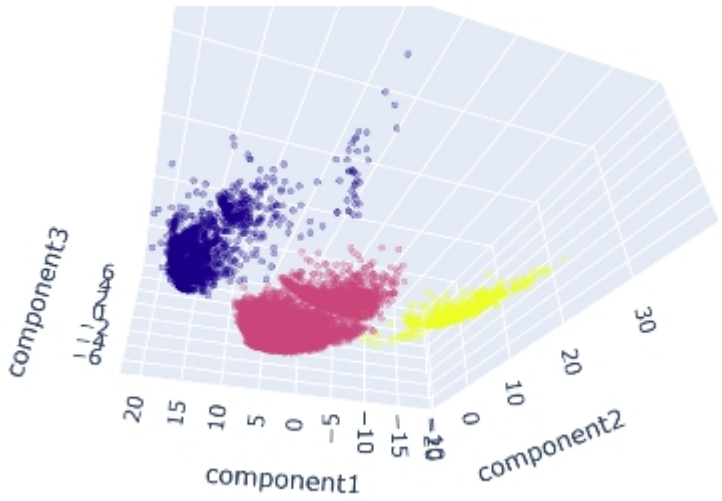


From visualization of first three PCA components of K-means algorithm I got a space of clear three clusters
That is a success

In [125…

```
Image(filename='k-means_three_clusters.jpg')
```

Out[125…

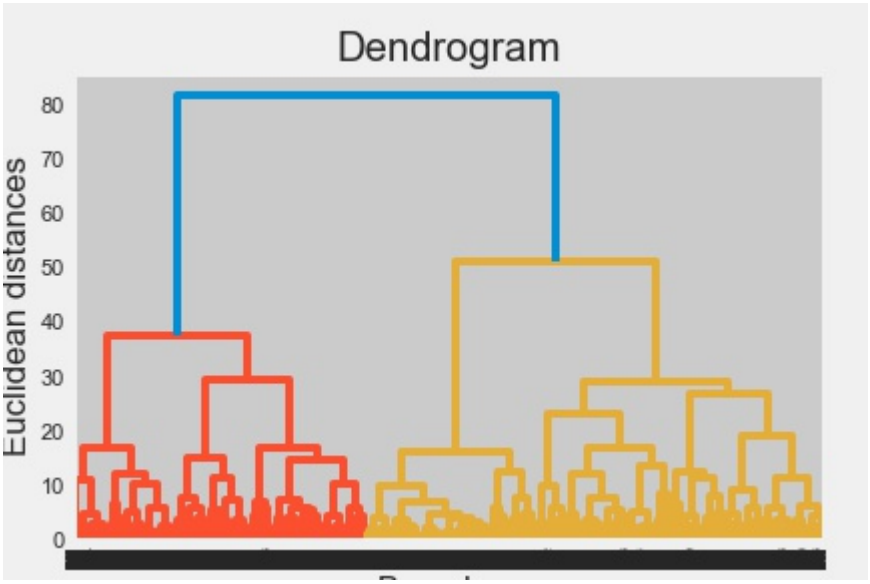


Dendrogram:

However a dendrogram trained on method = 'ward' suggests two clusters:

In [127…
```
Image(filename='dendrogram.jpg')
```

Out[127…



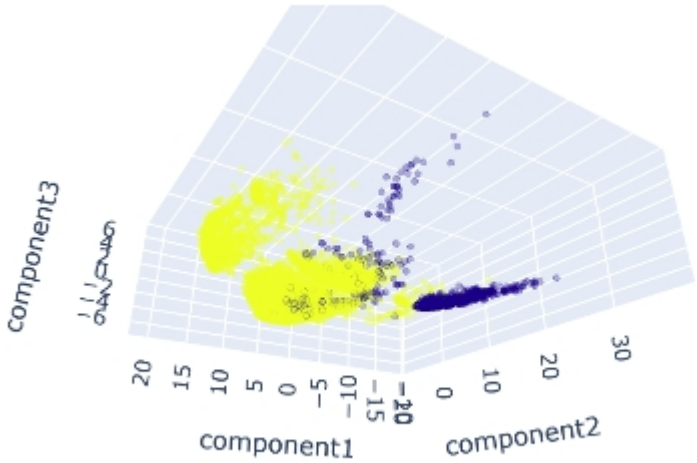
DBSCAN:

Regardless of selected parameters for DBSCAN algorithm the clustering returns either one or two clusters

In below case I have had used: eps=0.4, min_samples=4, metric="kulsinski"

In [128…
```
Image(filename='dbscan_two_clusters.jpg')
```

Out[128…



More details on GitHub

5) A paragraph explaining which of your Unsupervised Learning models you recommend as a final model that best fits your needs in terms.

From the 3-D scatter plot perspective it looks like K-means clustering algorithm done a really good job on a matter of finding seperate clusters of credit cards clients.

This is the model I recommend at the moment

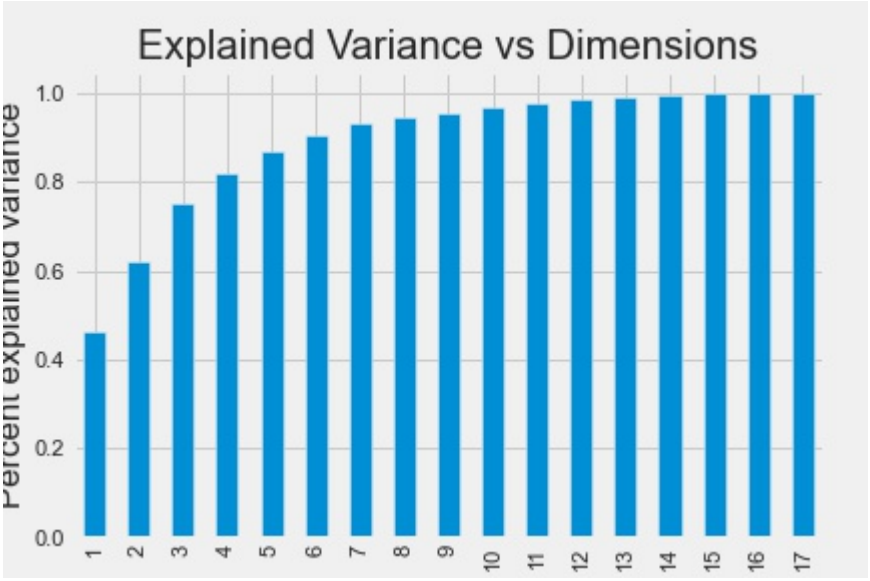
More details on GitHub

6) Summary Key Findings and Insights, which walks your reader through the main findings of your modeling exercise.

Taking under the regard a fact that first three components are expaining almost 80% of data set's variance

In [130…
```
Image(filename="explained_variance_vs_dimensions.jpg")
```

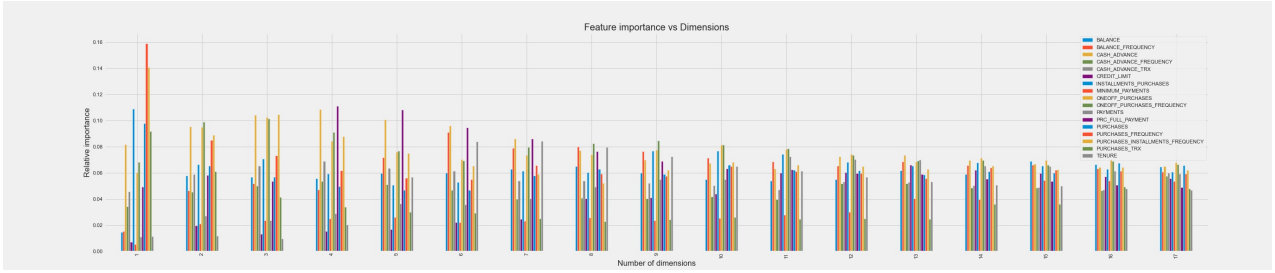
Out[130…



From Feature Importance for a number of dimensions "3" we can asssume that...

In [131…
```
Image(filename="feature_importance_vs_dimensions.jpg")
```

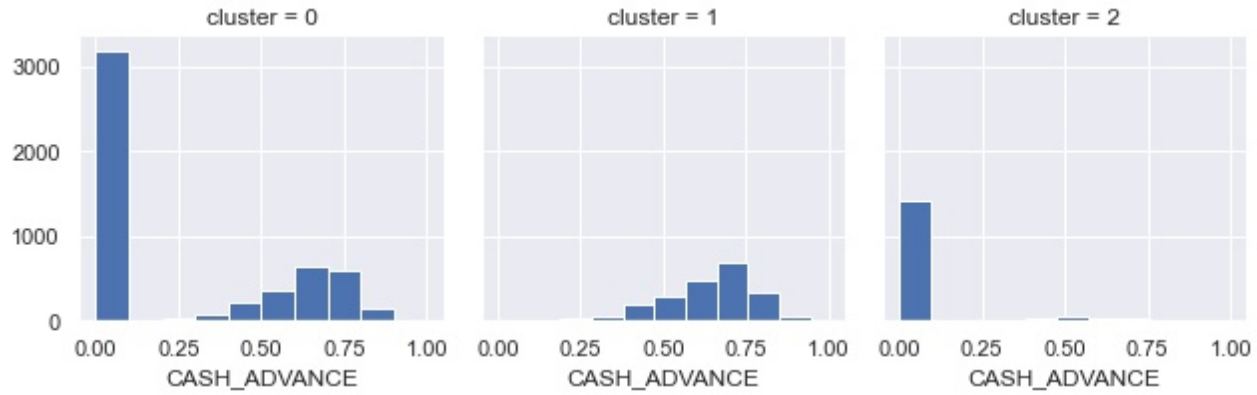
Out[131…



- CASH_ADVANCE

In [133…
```
Image(filename="three_clusters_histograms_of_CASH_ADVANCE.jpg")
```

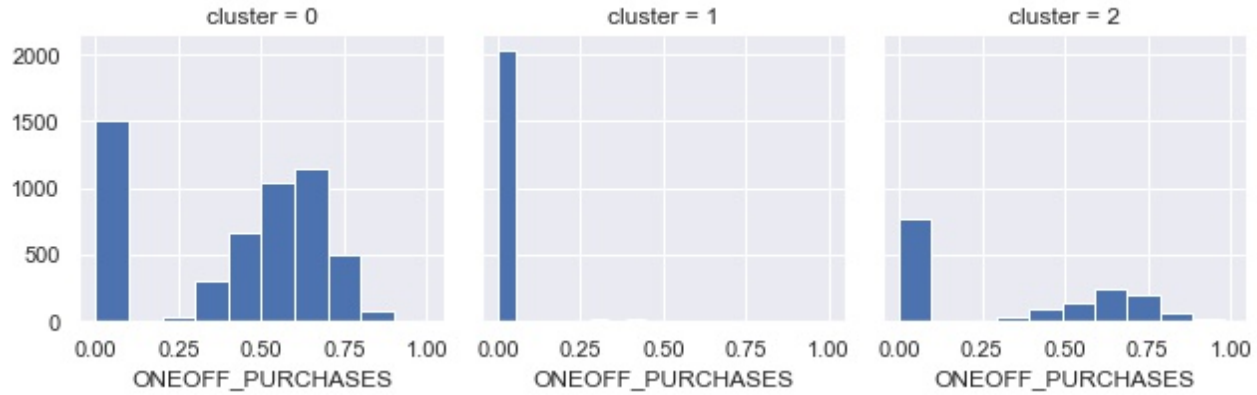
Out[133…



- ONEOFF_PURCHASES

In [135…
```
Image(filename="three_clusters_histograms_of_ONEOFF_PURCHASES.jpg")
```

Out[135…



- PURCHASES_FREQUENCY

In [136…
```
Image(filename="three_clusters_histograms_of_PURCHASES_FREQUENCY.jpg")
```

Out[136…

and

- PURCHASES_INSTALLMENTS_FREQUENCY

In [137…
```
Image(filename="three_clusters_histograms_of_PURCHASES_INSTALLMENTS_FREQUENCY.jpg")
```

Out[137…

are the most important variables on the matter of explaining the variance of data set.

We should now try to name each one of clusters in a regard to what kind of characteristics it has revealed to us.
Like:

- cluster = 0 as Big spenders
- cluster = 2 as Medium spenders
- cluster = 1 as No spenders

More details on GitHub

7) Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model or adding specific data features to achieve a better model.

As additional steps I would suggest:

- More feature engineering like adding polynomials
- Testing if the results of clustering are right on a hold out data set