Identify Customer Segments

REVIEW

HISTORY

Meets Specifications

Dear Student,

Congrats on completing your Identify Customer Segments project!

You seem to have an excellent understanding of the topic.

I wish you good luck with the next project and all the best in your future :)

P.S.
Take a look at this awesome library for data visualization and clustering:
https://hypertools.readthedocs.io/en/latest/

One more thing :)
Please rate my work as project reviewer! Your feedback is very helpful and appreciated.
Thank you so much!

Preprocessing

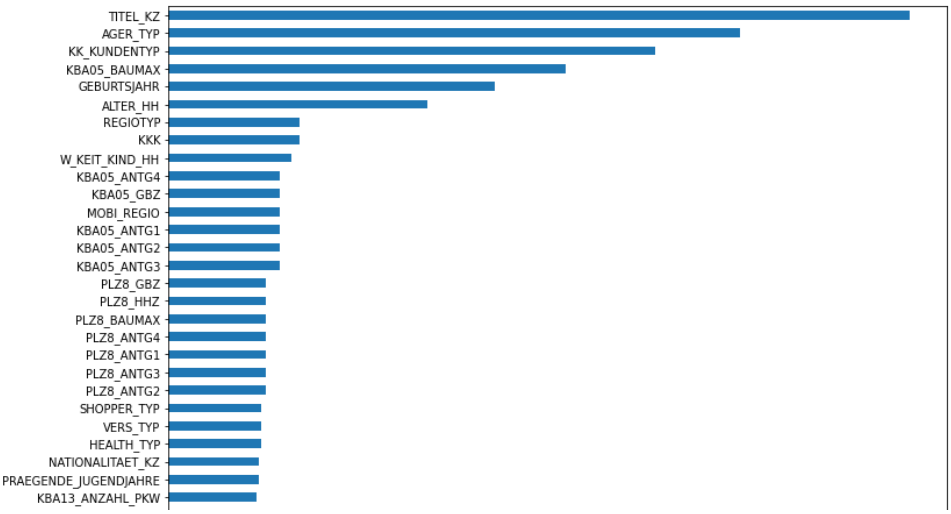
All missing values have been re-encoded in a consistent way as NaNs.

Excellent job using feat_info to encode missing values into NANs.

Columns with a large amount of missing values have been removed from the analysis. Patterns in missing values have been identified between other columns.

Good work finding and removing columns that have a lot of missing values.

I would have removed GEBURTSJAHR AND ALTER_HH as well.




```
# Remove the outlier columns from the dataset. (You'll perform other data
# engineering tasks such as re-encoding and imputation later.)

dropped_columns = ["TITEL_KZ", "AGER_TYP", "KK_KUNDENTYP", "KBA05_BAUMAX"]
azdias_cols_dropped = azdias.drop(dropped_columns, axis = 1)
azdias_cols_dropped.head()
```

Mixed-type features have been explored, resulting in re-engineered features.

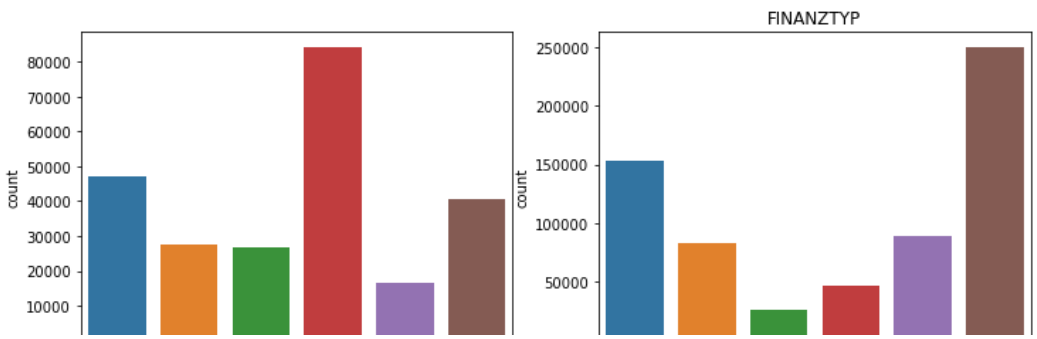
Excellent job re-engineering the mixed-type features.

```
# Set decade
one_hot_encoded_data["PRAEGENDE_JUGENDJAHRE_DECADE"] = one_hot_encoded_data["PRAEGENDE_JUGENDJAHRE"].replace(range(1, 16), [40, 40, 50, 50, 60, 60, 60, 70, 70, 80, 80, 80, 80, 90, 90])
```

```
# Set movement, 1 for Mainstream, 2 for Avantgarde
one_hot_encoded_data["PRAEGENDE_JUGENDJAHRE_MOVEMENT"] = one_hot_encoded_data["PRAEGENDE_JUGENDJAHRE"].replace(range(1, 16), [1, 2, 1, 2, 1, 2, 2, 1, 2, 1, 2, 1, 2, 1, 2])
```

The data has been split into two parts based on how much data is missing from each row. The subsets have been compared to see if they are qualitatively different from one another.

Good work splitting the dataset into two different parts based on the number of missing values in each row, and comparing these parts.



Rows with more missing values

Rows with less missing values

Categorical features have been explored and handled based on if they are binary or multi-level.

Excellent work handling the binary and multi-level categorical features.

Dataset includes all original features with appropriate data types and re-engineered features. Features that are not formatted for further analysis have been excluded.

Good work dropping the columns that will not be used in further analysis.

A function applying pre-processing operations has been created, so that the same steps can be applied to the general and customer demographics alike.

Excellent work writing the cleaning function :)

Feature Transformation

Feature scaling has been properly applied to the demographics data. Imputation has been performed to remove remaining missing values.

Nice work imputing and scaling your data.

Weights on at least three principal components are used to make inferences on correlations between original features of the data. General meanings are ascribed to principal components where applicable.

Excellent job interpreting the PCA components and inferring correlation between original features.

Principal component analysis has been applied to the data to create transformed features. A variability analysis has been performed to justify a decision on the number of features to retain.

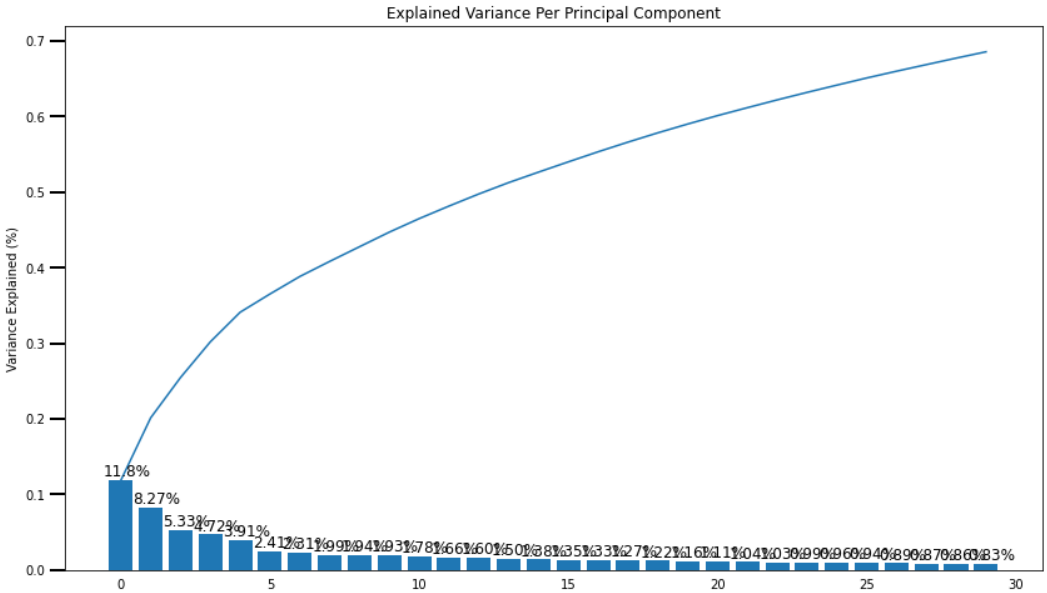
Good word choosing a reasonable number of feature so to explain around 70% of variance in the data. This is on the low end one usually goes for 90%.

Principal Component

```
# Re-apply PCA to the data while selecting for number of components to retain.

pca = PCA(30)
pca.fit(scaled_data)
X_pca = pca.transform(scaled_data)
```

```
scree_plot(pca)
```



Clustering

Multiple cluster counts have been tested on the general demographics data, and the average point-centroid distances have been reported. A decision on the number of clusters to use is made and justified.

Good work choosing the right number of clusters. You did not have to compute the SSE yourself you could have used the score method of the model. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.score

Cleaning, feature transformation, dimensionality reduction, and clustering models are applied properly to the customer demographics data.

Well done. Re-using same fitted transformers on demographic dataset and applying them to customer dataset, allows us to make direct comparisons between the clusters created with the two different dataset.

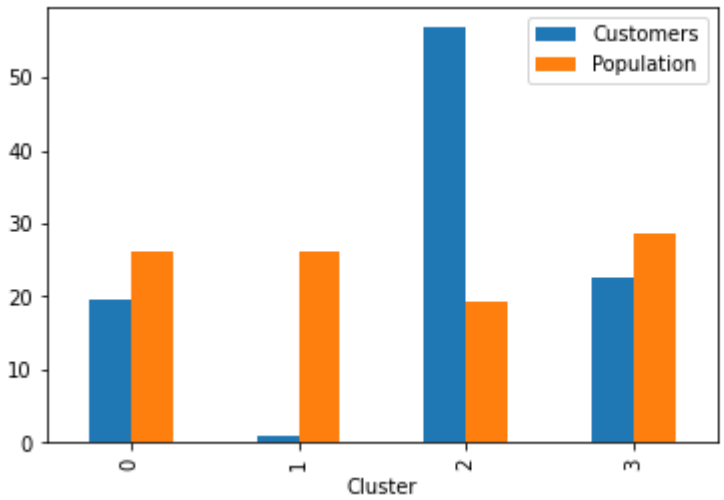
```
customers_with_imputer = imputer.transform(customers_dropped_na)
customers_scaled_data = scaler.transform(customers_with_imputer)
```

```
X_cust = pca.transform(customers_scaled_data)
```

```
cust_clusters = kmeans.predict(X_cust)
```

A comparison is made between the general population and customers to identify segments of the population that are central to the sales company's base as well as those that are not.

Excellent work identifying clusters in which customers are overrepresented/underrepresented.



DOWNLOAD PROJECT

RETURN TO PATH