UDACITY

Return to "Data Scientist Nanodegree" in the classroom

Identify Customer Segments

REVIEW

HISTORY

Requires Changes

10 SPECIFICATIONS REQUIRE CHANGES

Your coding skills is quite impressive. You just need to fine tune some of these sections and answer the final ones and you will be good to go, but should be simple fixes and great for learning the material even better. Keep up the great work!!

Preprocessing

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

Missing value codes given in feat_info's last column have been used to convert all codes to NaNs. Nicely done!

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

Nice visuals to support your removal of these 6 features. These features definitely are outlier NaN columns.

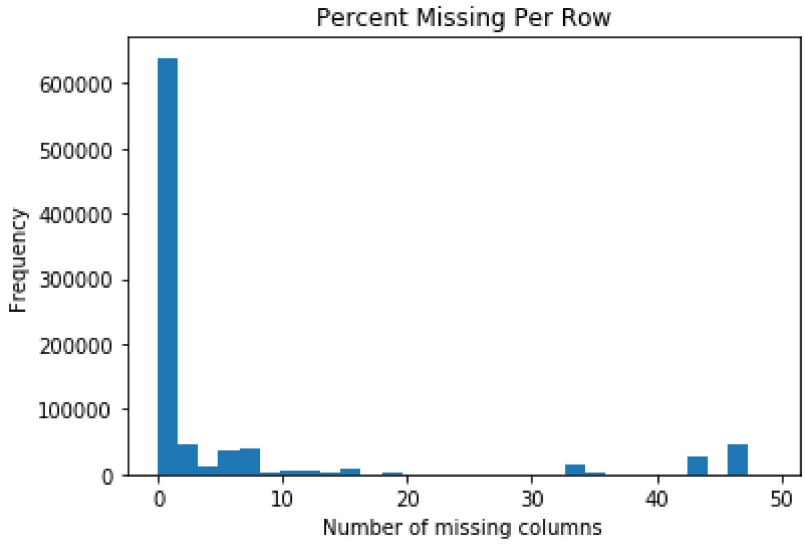
In terms of missing value patterns

- the “FINANZ” and “SEMIO” feature sets are both complete, lacking any missing values. All features that do not have missing values are on the person level of recording. This is very typically to see with demographic data.

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.

```
if value < 3: # lowest bound is 3%
```

A threshold value of 3 is a tad low. If you were to plot the number of missing values per row, you would get



Thus we can see that we have large open gap between 9 and 32 missing values, you should choose a value within that range.

GOOD

Visuals are quite nice here. You might also look into plotting some seaborn distplots. For example.

```
plt.figure(figsize=(100,100))
for i, col in enumerate(azdias.columns[:10]):
    plt.subplot(5, 2, i+1)
    sns.distplot(azdias_below[col][azdias_below[col].notnull()], label='below')
    sns.distplot(azdias_above[col][azdias_above[col].notnull()], label='above')
    plt.title('Distribution for column: {}'.format(col))
    plt.legend();
```

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

For the `OST_WEST_KZ` feature, instead of one hot encoding this feature (and creating an unnecessary column), you should encode the `OST_WEST_KZ` feature numerically (e.g. as 0, 1 instead of 'O', 'W' or vice versa).

For example

```
azdias['OST_WEST_KZ'] = azdias['OST_WEST_KZ'].apply(lambda x: 0 if x == 'O' else 1)
```

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

> "All mixed type features were kept"

If you want to use either the `WOHNLAGE` or `PLZ8_BAUMAX` features, you are going to need to perform some additional steps.

WOHNLAGE

- If you want to use the information in this variable, you will need to decide how to encode 7 and 8 relative to the rest of the ordinal scale (such as treating them as missing values, to later be imputed at the mean), as well as decide whether or not you should engineer a second variable that acts as a rural flag (e.g. 1-5 = not rural, 7-8 = rural).

PLZ8_BAUMAX

- This variable is analogous to KBA05_BAUMAX, which notes the most common building type within the region. The scale is ordinal from 1-4 with size of housing buildings, but 5 indicates business buildings. Like KBA05_BAUMAX, it could be argued that the feature could be dropped (as noted in Step 1.1.2), but if you want to keep this feature you need to indicate what to do with the level-5 values. This might include just making a domain assumption that business buildings would be of a density on the same or higher level than level-4 values (10+ family homes).

You could also either drop or simply dummy encode these two 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.

You might look into creating a couple new features for the `WOHNLAGE` feature. Maybe engineer a new variable that acts as a rural flag (e.g. 1-5 = not rural, 7-8 = rural).

Here might be a couple of function ideas to explore for the `WOHNLAGE` feature.

```
def create_WOHNLAGE_neigh(row):
    if np.isnan(row): return row
    if row in [1., 2., 3., 4., 5.]: return row
    else: return 0
```

```
def create_WOHNLAGE_rural(row):
    if np.isnan(row): return row
```

```
    if row in [7., 8.]: return row
    else: return 0
```

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

For this `clean_data()` and `RemoveNanColumns()` function, Instead of checking with `if ratio_loc > 20:` and potentially dropping *different* columns, you should drop the **same** 6 columns that you did from your demographic dataset of `AGER_TYP, GEBURTSJAHR, TITEL_KZ, ALTER_HH, KK_KUNDENTYP and KBA05_BAUMAX` (since your customer dataset and demographic dataset should match in terms of columns).

Feature Transformation

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

Discussion 2.1: Apply Feature Scaling

(Double-click this cell and replace this text with your own text, reporting your decisions regarding feature scaling.)

Nice use of StandardScaler, however to meet specifications for this rubric point, you should provide some justification for your decisions here. Why do we need to scale the data?

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.

Discussion 2.2: Perform Dimensionality Reduction

(Double-click this cell and replace this text with your own text, reporting your findings and decisions regarding dimensionality reduction. How many principal components / transformed features are you retaining for the next step of the analysis?)

This section is not answered. Why was `n_components=50` chosen?

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.

This section is not answered.

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.

This section is not answered.

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

This section is not answered.

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.

This section is not answered.

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)

RETURN TO PATH

Rate this review