UDACITY

‹ Return to Classroom

Identify Customer Segments

REVISÃO

HISTORY

Meets Specifications

Dear student

Great job on your project! You've clearly understood the material from the tutorials and you've done a great job applying it to this real-world dataset. Congratulations on finishing up quickly and good luck with the next section of the course.

Cheers!

Preprocessing

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

> It seems, from the histogram, that the NaN's above a threshold of 200000 are uniformly distributed and they seem to be kind of outliers, therefore they have been removed.

You've done a great job identifying the columns that are outliers in terms of high missing values. The columns that you've removed are definitely outliers in terms of missing values.

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

```
for feat in range(feat_info.shape[0]):
#for featttt in range(1):
#    feat=57
    missings=feat_info['missing_or_unknown'][feat].split('[')[1].split(']'
)[0].split(',')

    misslist=[]
    misslist2=[]
#    print(missings,len(missings))
    for i in range(len(missings)):
#        if( isinstance(int(missings[i]),int)):
        if( missings[i]!='' and missings[i]!='X' and missings[i]!='XX'):
            misslist.append(int(missings[i]))
        else:
            misslist.append(missings[i])
#    print(misslist)
#            continue
#    if(misslist!=[]):
    colname=listcols[feat]
#    print(colname)
    azdias_clean[colname].replace(misslist,np.nan,inplace=True)
```

Nice work removing the missing value codes and replacing them with NaN's! You've done a great job handling the `X` and `XX` special cases.

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

Nice work generating the new features. Very creative work in flipping around the values of the `CAMEO_INTL_2015` feature for easier interpretability.

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

> I used pd.get_dummies () because it is a simple and straightforward technique to handle categorical features.

Looks good! I think you've done a good job balancing the preprocessing requirements with the issue of dimensionality. It's certainly important to avoid diluting the variability captured by components in the PCA step.

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.

Nice job! I think you've done a good job choosing a reasonable threshold. There are two clear clusters in the

Nice job! I think you've done a good job choosing a reasonable threshold. There are two clear clusters in the rows with high missing value counts: one at < 9 missing values per row and another at >30 missing values per row. Either of these (or anything in between) is a good cutoff point for this 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.

```
newdf=clean_data(azdias)
```

Very nice! Great employment of all your work into the clean_data() function. I'd also commend you for testing the function out in order to verify that it works on the general demographics data.

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

```
df_clean2.drop(['PRAEGENDE_JUGENDJAHRE'],axis=1,inplace=True)
df_clean2.drop(['CAMEO_INTL_2015'],axis=1,inplace=True)
```

Looks good!

Feature Transformation

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

Great job describing your preprocessing methodology! I'd also suggest trying to impute the missing values using "most frequent" or "median" methods. This might show some interesting patterns when you analyze the clusters in your downstream analysis.

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.

Nice analysis of the top 3 principal components!

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.

> I am aiming to retain 80% of the total information by using PCA. Therefore, as the above analysis

> I am aiming to retain 80% of the total information by using PCA. Therefore, as the above analysis has shown, I am using 75 components, which is quite a significant dimensionality reduction!

Great choice! I think ~80 PC's is right in the sweet spot. You could definitely have dropped or kept a few more components, but retaining most of the variance in the dataset while reducing the dimensionality of the dataset by 2/3 is pretty optimal.

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.

> By using the Elbow method of KMeans, I have found an (nearly) optimal number of clusters as to be 21.

Looks good! There definitely wasn't a distinct elbow in either of the curves so this is a bit of a judgement call. Anywhere from 15-21 clusters looks pretty good.

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.

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

```
pred_opt_cust=km_opt.predict(custdf2_scaled_pca)
```

Looks good! By re-using the same methods you've already fit, you can really tell how this feature engineering and unsupervised learning will look when you apply them to independent data in the future.

BAIXAR PROJETO

RETORNAR

Avalie esta revisão

Avalie esta revisão

COMEÇAR