

# Machine Learning Engineer Nanodegree: Capstone Report

## Arvato Solutions: Customer Targeting

Author: Mike Griffin

Date: 26<sup>th</sup> May 2020

### CONTENTS

- 1) Domain Background
- 2) Problem Statement
- 3) Datasets & inputs
- 4) Solution statement
- 5) Report - customer segmentation
  - a. Data processing & feature engineering
  - b. Principal component analysis
  - c. Clustering analysis
- 6) Report - customer acquisition
  - a. Benchmark modelling
  - b. Model training and selection
  - c. Model results
  - d. Kaggle submission
  - e. Model interpretation
- 7) Summary – conclusions and further work
- 8) References

## DOMAIN BACKGROUND

This is a Udacity capstone project representing a real-life data science problem using information provided by Betrelsmann Arvato Analytics.

Arvato is a global services company that develops and implements solutions for business customers in a diverse range of industries, with a focus on innovative data analysis and automation. For this project, Arvato provided services to a German mail-order sales company aiming to understand demographics and improve customer acquisition. This project will explore unsupervised clustering to identify customer groups as well as classification modelling to predict which individuals will respond to campaigns.

Ultimately this work was designed allow the client to improve client targeting for future campaigns. My analysis is for exploration and learning only.

## PROBLEM STATEMENT

The overarching objective is to identify which individuals are most likely to respond to a marketing campaign and become customers of the client. This is a binary classification problem using individual-level attributes to estimate the likelihood of a customer response. Specifically, the project will aim to maximise the ROC score on a held-out test set on Kaggle.

## DATASETS AND INPUTS

Demographics information has been provided for both the general population at large as well as for prior customers of the mail-order company, stored at the level of an individual.

- Udacity\_AZDIAS\_052018.csv: Demographics data for the general population of Germany; 891 211 persons (rows) x 366 features (columns).
- Udacity\_CUSTOMERS\_052018.csv: Demographics data for customers of a mail-order company; 191 652 persons (rows) x 369 features (columns).
- Udacity\_MAILOUT\_052018\_TRAIN.csv: Demographics data for individuals who were targets of a marketing campaign; 42 982 persons (rows) x 367 (columns).
- Udacity\_MAILOUT\_052018\_TEST.csv: Demographics data for individuals who were targets of a marketing campaign; 42 833 persons (rows) x 366 (columns).

365 fields are common across the dataset with information on the fields and labels stored in two data dictionaries:

- DIAS Information Levels - Attributes 2017.xlsx: high-level descriptions of attributes
- DIAS Attributes - Values 2017.xlsx: detailed mappings of levels within fields

## SOLUTION STATEMENT

The solution will have two components:

- 1) **Customer segmentation**— a report which will separate the population in to distinct clusters and explore the variation in attributes by clusters. This will require appropriate imputation, feature engineering, dimensionality reduction and clustering via k-means. This will allow comparison between the population and customer data to explore which clusters are over/under-represented in the customer set.
- 2) **Customer acquisition** - a set of models will be trained and compared on the binary classification of whether an individual will respond to a campaign. Segment information from (1) will be computed and fed in to the classification model. Model performance will be assessed and tuned using train/validation splits before measurement on a held-out test set on Kaggle.

## EVALUATION METRICS

For the first stage involving customer segmentation:

- Explained variance will be used to assess the effectiveness of dimensionality reduction.
- For clustering, the “elbow method” will be used to assess the optimal number of clusters using the inertia metric to minimise cluster distances.

For the second stage involving binary classification:

- In line with the Kaggle competition, the AUC ROC score will be the main measure to assess model performance, supplemented by analysis of confusion matrices.

## 1) REPORT- CUSTOMER SEGMENTATION

### a) Data processing and feature engineering

Examination of the target variable (“RESPONSE”) within the “mailout\_train” set highlights that the classification problem has highly imbalanced classes with successful responses in 1.3% of cases. This means over/under sampling will be required to achieve good results.

Basic stats on the datasets are collated below – it can be seen that fields are shared across most of the datasets. The customer dataset is c20% the size of the population dataset and the classification train and test sets are of similar size.

#### Input datasets:

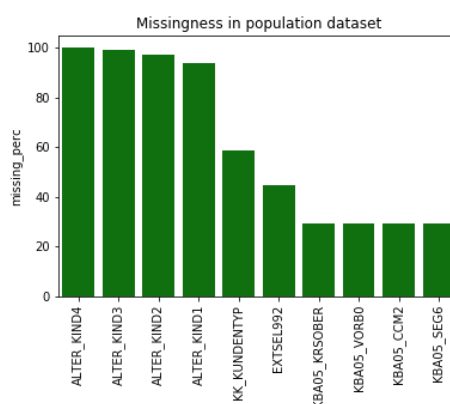
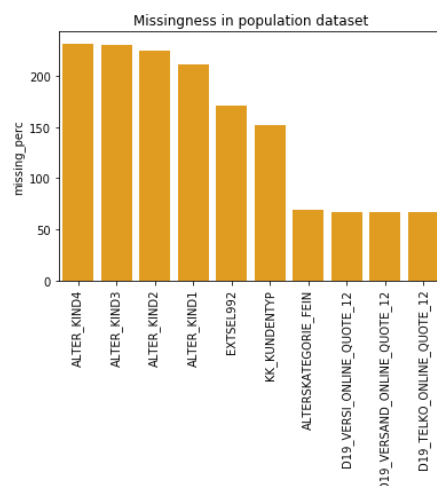
Dataset	Float columns	Integer columns	Objects	Rows	Size
Azdias	267	93	6	891,221	2.4GB
customers	267	94 (+??)	6	191,652	540MB
mailout_train	267	94 (+“RESPONSE”)	6	42,962	120MB
mailout_test	267	93	6	42,832	120MB

This highlights here are six categorical variables requiring specific treatment:

- EINGEFUEGT\_AM – this is a datetime field from which the year, month and day is extracted as separate features and the datetime feed is dropped
- CAMEO\_DEUG\_2015 and CAMEO\_INTL\_2015 - numerical 0 used to replace X or XX and all values converted to floats
- CAMEO\_DEU\_2015 and D19\_LETZTER\_KAUF\_BRANCHE – one hot-encoded (note that all dummies are included as tree models are used)
- OST\_WEST\_KZ – “W” or “E” binary encoded with gaps as -1

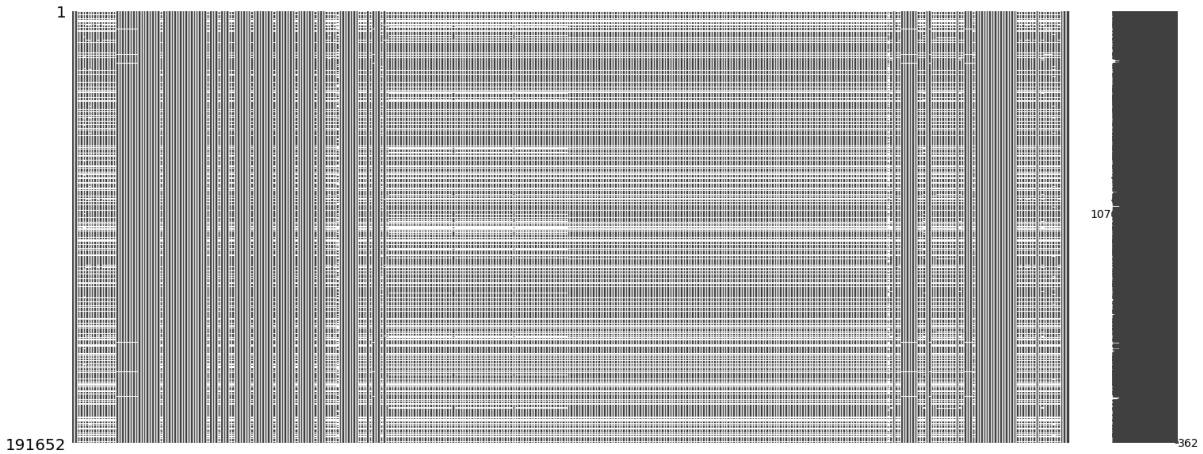
No fields were removed to allow for the possibility all feeds offer predictive power (inc “LNR”). The population dataset was randomly sampled by row (to include 50% data) to improve computation times.

Exploring patterns of top missingness suggests this is similar between the population and customer datasets



Using the missingno library visually highlights how the missingness uniform across sets of fields suggesting these originate from specific feeds. Imputation functions were used to replace gaps with median values, add binary flags for missingness and then compress these flags across data feed sets.

This imputation approach means that the distributions of the underlying fields are unchanged and the missingness itself can be explored for predictive power. Overall the data processing creates a dataset of 457 variables from the original set of 366 all of which are numerical and complete



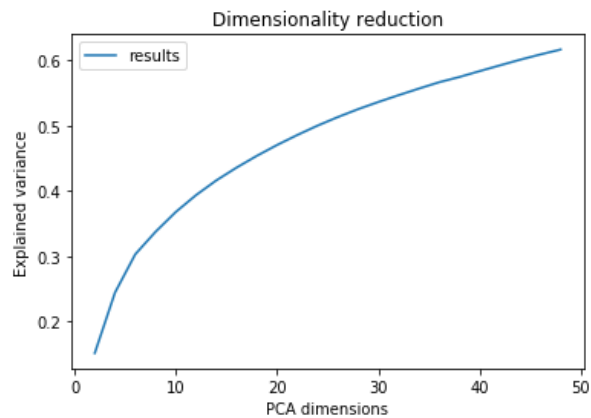
Data pipelines:

For the unsupervised and supervised tasks, two pipelines were used using common functions where appropriate.

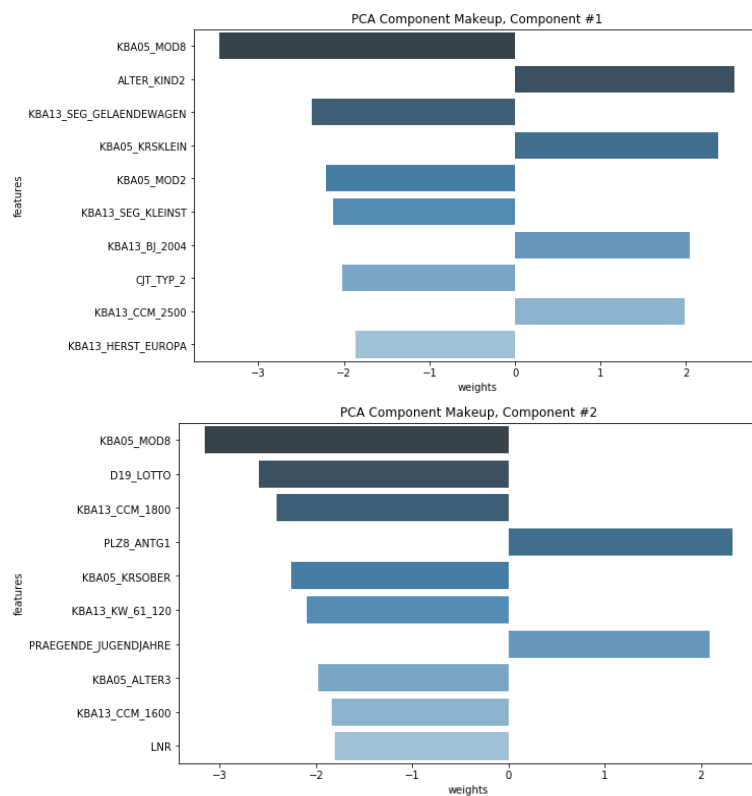
Step	Data prep	Approach	Unsupervised - PCA and clustering	Supervised - Binary Classification
1	Add binary flags for nas	Write as function	N	Y
2	Add date variables	Process_dates function to extract month, year	Y	Y
3	Categorical and gap-filling	Process_df function to impute median values and add encoding	Y	
4	Consolidate nas	clean_nas module to consolidate na flags across systematic data gaps	N	Y
5	Rescaling	Standard scaler trained on population dataset	Y	N
6	Principal component analysis	Via sklearn pca	Y	N
6	Clustering	Via sklearn k-means	Y	N (cluster added as variable)
7	Classification modelling	Sklearn tree models and logistic regression	N	Y

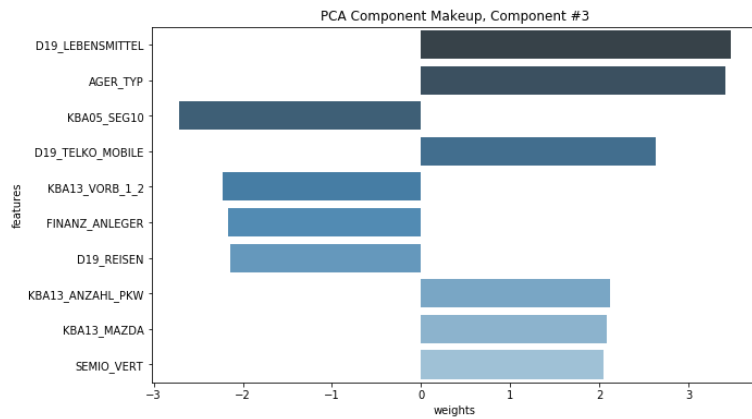
## b) Principal Component Analysis

To improve clustering results, the 300+ dimension dataset was first compressed via principal component analysis. PCA was run on the underlying dataset with median imputation to avoid dimension expansion due to missingness. A PCA model was fitted using default parameters with exploration of dimension in range of 1 – 100.



Using 100 dimensions was found to capture 76% of variance. Attributes of top PCA components are illustrated below although these do not appear to have common-sense interpretations.





Analysing the principal components shows the attributes associated

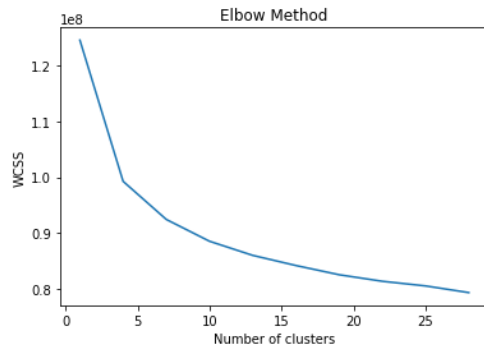
To demonstrate a few examples with more obvious interpretation

- PCA 1: Low share of low power cars, high share of powerful cars, blank household income, low share of old cars,
- PCA 2:
- PCA 3: money savers, low financial prep, traditional mindset, older individuals

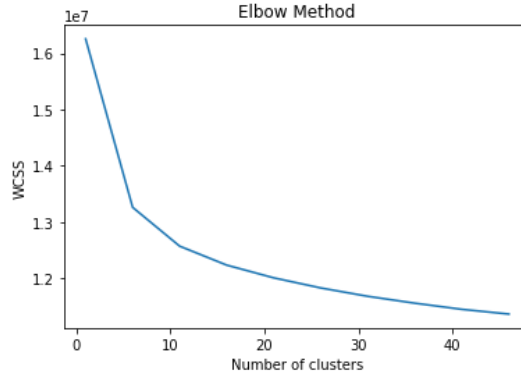
### c) Clustering

K-means clustering was run on the full and PCA spaces to judge the most appropriate dimension adjustment. As expected, cluster differences are seen to be smaller in PCA-space

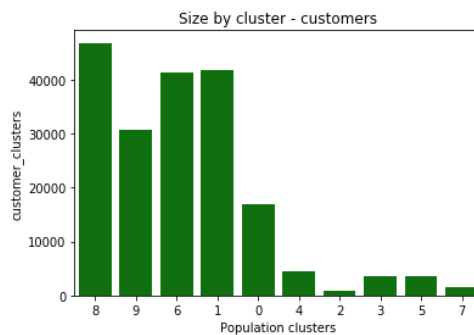
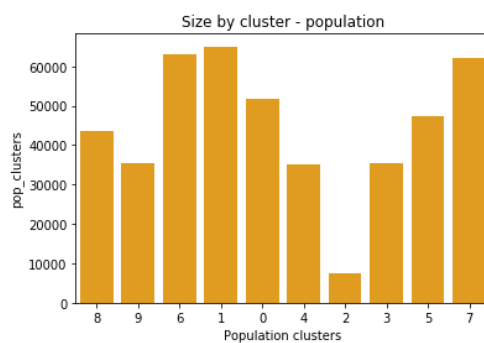
Clustering in PCA space:



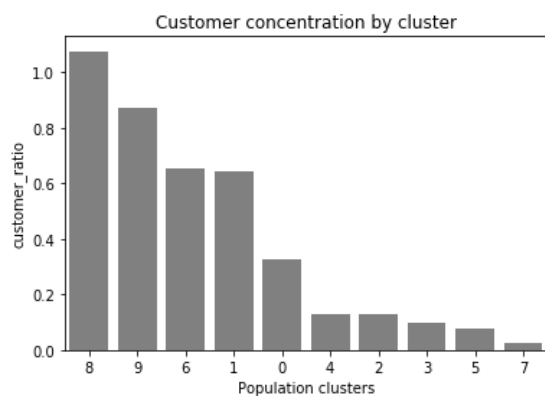
Clustering in full space



This chart suggests an inflection in the range of 8-10. Using 10 clusters yields the below results when applied across the population ad customer base



Focussing on the ratio of customer cluster size to population indicates the representation of groups in the customer dataset vs the population. Note that the ratio exceeds 1 for group 8; this is not necessarily a surprise given that this analysis is based on 50% of the population dataset for computational efficiency.



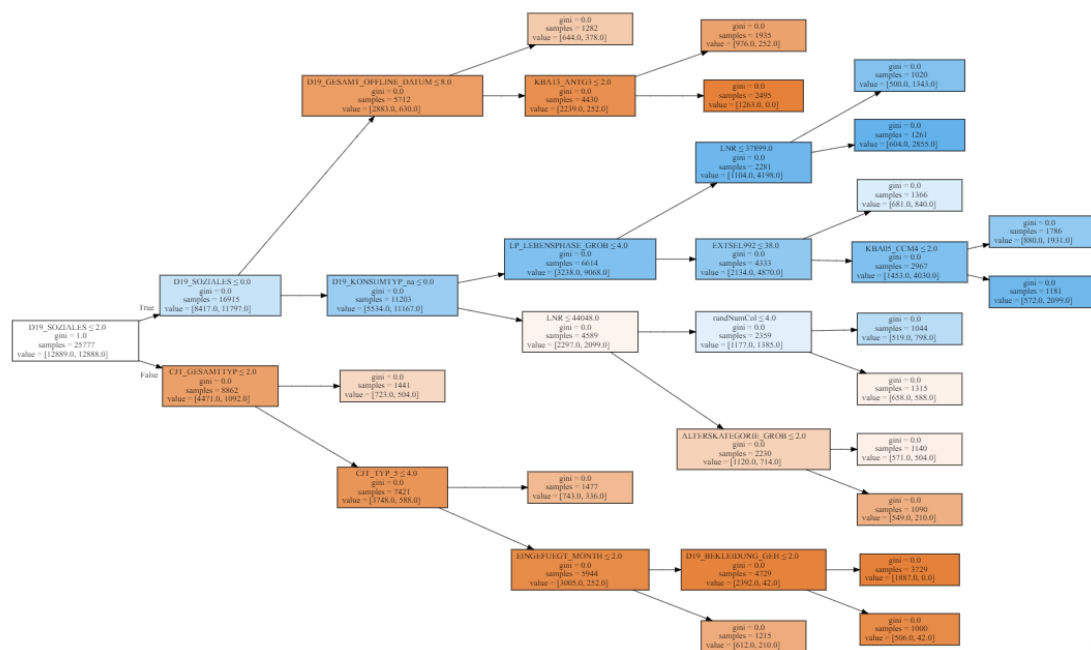


## REPORT- CUSTOMER ACQUISITION

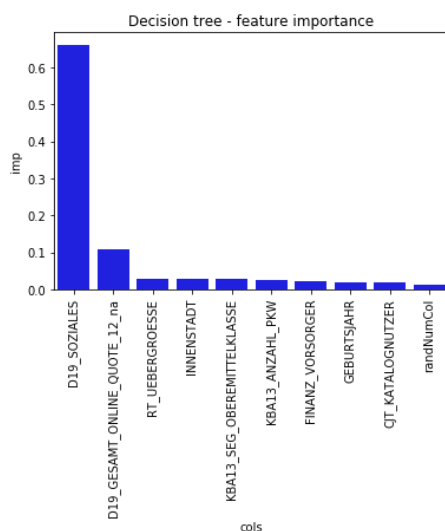
### a) Benchmark modelling

BENCHMARK MODEL – A simple decision tree will be trained to determine baseline performance, with constraints on leaf size or depth to control overfitting. Baseline performance will be assessed using the ROC AUC score and confusion matrices. This will enable clear interpretation of important metrics, thresholds and interactions which may be used to engineer features with predictive value for the main modelling.

Trained decision tree classifier with balanced class weights and minimum samples leaf of 1000, otherwise default parameters

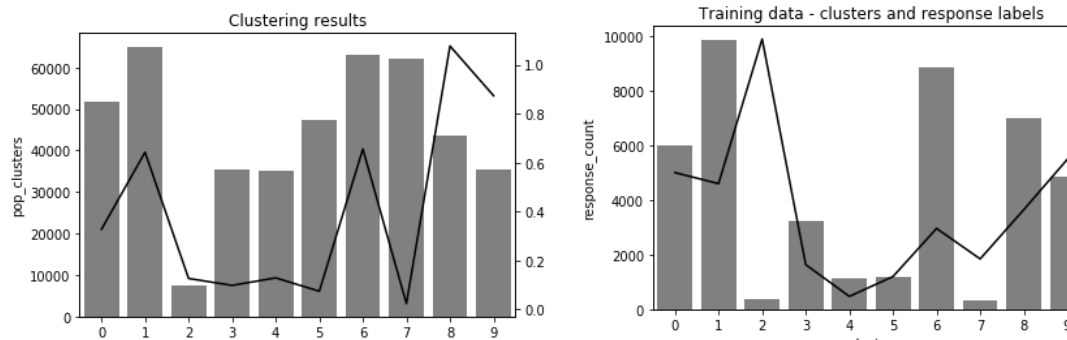


Using feature importance demonstrates that the predictive power is dominated by the variable “D19\_SOZIALES”



## Using clustering data

To explore the usefulness of the cluster information, I explored the response rate by group. If this clustering from task 1 is robust and predictive then clusters which are over-represented in the customer/population analysis would be expected to show higher response rates. Visually this does not appear to be the case



Using a random forest with just cluster information as input yields a low score of  $\sim 0.5$  indicating that the cluster information provides no predictive information itself. Task 1 should be revisited to explore whether improvements can be made.

## b) Model training and selection

No structure is evident in the datasets (eg time series) so random partitions are used to model training, validation and testing. A specific held-out validation set was created to compare models and submissions to Kaggle were intentionally limited to avoid overfitting to the test set.

Dataset	Approach	Design
Training & tuning	For hyperparameter tuning	Five-fold cross validation using grid search
Validation	For comparison of model outcomes	25% held out
Final test	Kaggle testing data	Using optimal hyperparameters on combined train and validation sets

Several classical ML model types were explored:

Model	Tuning approach	Comments
Random forest	Grid search: <ul style="list-style-type: none"><li>• class_weight = balanced / balanced_subsample</li><li>• 'min_samples_leaf' varied in range [2,2000]</li><li>• 'max_features' varied in range [&lt;0.1,1]</li></ul> Optimal performance found with 1200 leaves and max_features=0.6	<ul style="list-style-type: none"><li>• Aggregation of decision trees with bagging to control overfitting</li><li>• Interpret via feature importance</li><li>• Also utilised random column to determine threshold for feature importance to define condensed models</li></ul>
Gradient boosting	Tuning of macro parameters: <ul style="list-style-type: none"><li>• 'N_estimators' in range [40,70]</li><li>• Max_depth in range [5,10]</li><li>• Subsample in range [0.5,1]</li></ul> Then tree-specific parameters: <ul style="list-style-type: none"><li>• 'min_samples_leaf' varied in range [2,2000]</li><li>• 'max_features' varied in range [&lt;0.1,1]</li></ul> Optimal performance found with 15000 leaves, max depth of 8, 40 trees, subsample and max_features of 0.7	<ul style="list-style-type: none"><li>• Sequential tree modelling</li><li>• Interpret via feature importance</li></ul>
Logistic regression	Simple approach using same training set as tree models	<ul style="list-style-type: none"><li>• Only used for simple benchmark. Limited performance and understanding as all features have not been encoded or explicitly ordered</li></ul>
Stacked models	Linear (parallel) combinations of above models. Combining models based on simple heuristics rather than full optimisation	<ul style="list-style-type: none"><li>• Offer best performance</li></ul>

### c) Model results

Results are reported below:

Model	Train ROC	Validation ROC	Kaggle test ROC	Notes
Benchmark – decision tree	0.815	0.758		
Logistic regression	0.875	0.620		
Random Forest	0.821	0.781	0.789	
Gradient Boosting	0.807	0.768	0.787	
Stacked model			0.7998	50:50 weighting between RF and GBM


Confusion matrix for stacked model

### d) Kaggle submission

Used simple model stacking with 50:50 weighting between gradient boosting and random forest predictions

57


MGriffin



0.79986

12

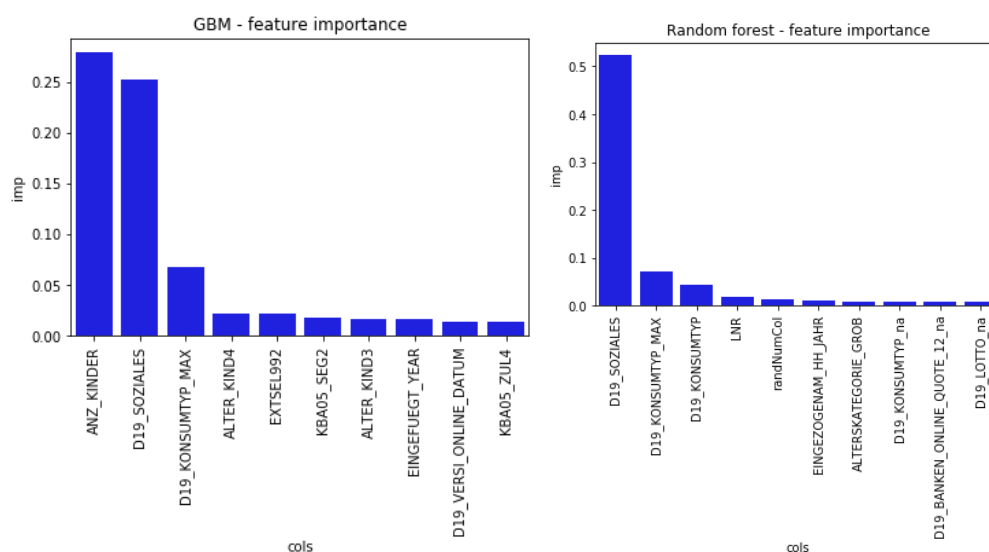
~10s

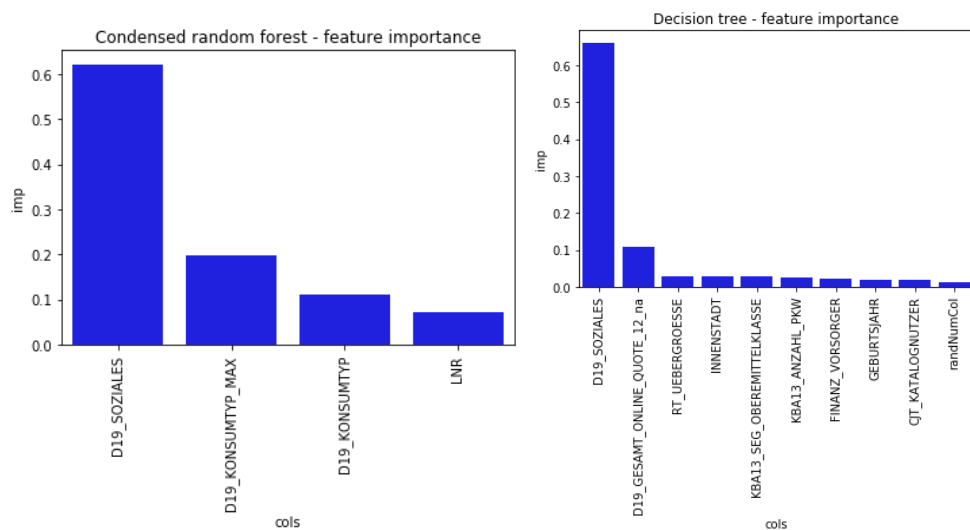
Your Best Entry 

Your submission scored 0.79974, which is not an improvement of your best score. Keep trying!

### e) Model interpretation

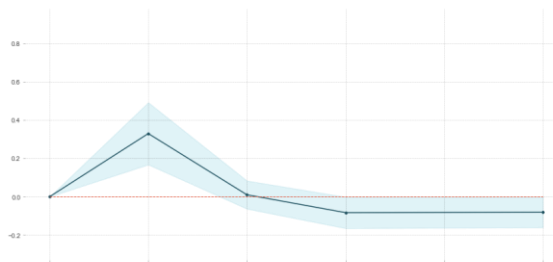
- In most of the models the most important feature is D19\_SOZIALES
- D19\_KONSUMPTION and D19\_KONSUMPTION\_MAX also feature as important
- ANZ\_KINDER



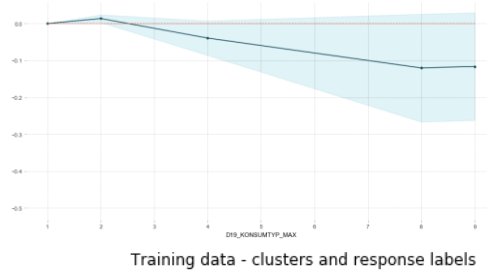


Focusing on the condensed random forest model which uses only features above the randomised column yields the below partial dependence charts

PDP for feature "D19\_SOZIALES"  
Number of unique grid points: 5

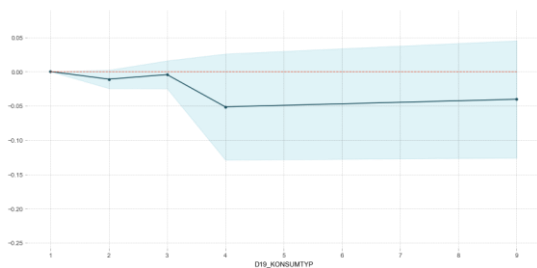


PDP for feature "D19\_KONSUMTYP\_MAX"  
Number of unique grid points: 5

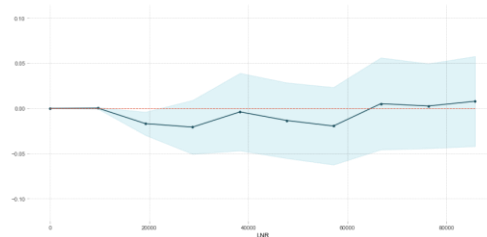


Training data - clusters and response labels

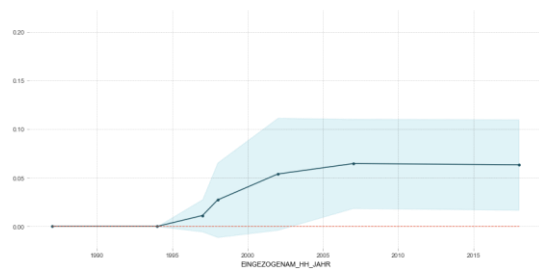
PDP for feature "D19\_KONSUMTYP"  
Number of unique grid points: 5



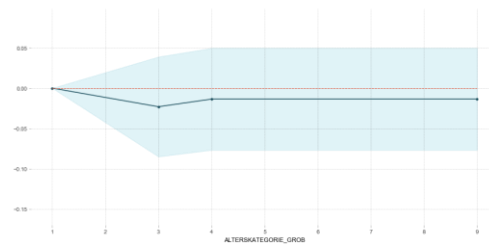
PDP for feature "LNR"  
Number of unique grid points: 10



PDP for feature "EINGEZOGENAM\_HH\_JAHR"  
Number of unique grid points: 7



PDP for feature "ALTERSKATEGORIE\_GROS"  
Number of unique grid points: 9



## SUMMARY

### Conclusions

This work explores customer clustering and predictive modelling.

Clustering was found to be difficult, likely due to high-dimensionality and missingness. The result is somewhat dissatisfactory and I intend to revisit the analysis in future.

Despite this, the core classification task was successful with a top~25% result on Kaggle via a standardised data science pipeline. Using simple combinations of tree models with basic

### Further Work

There are a number of additional tasks I would like to explore further:

- Revisiting PCA and clustering to produce groupings which offer intuitive insights and predictive power with classification task
- Testing other imputation approaches on key variables – current approach just using median to avoid shifting distribution
- Explore XGBoost rather than sci-kit learn implementation of gradient boosting
- SMOTE for more-sophisticated sample rebalancing to improve classification results or suing imblearn package
- Test auto-ML results for optimised model stacking