Colruyt Group

Look-alike Modeling for Highbrow Wines

May 08, 2019

Arif Thayal

Agenda

« Data Science Solution »

Context and Briefing

Data Discovery

Data Preparation

Modeling Approach

Decision Making

Deliverables

Context & Briefing

Problem Statement: To identify the customers who are most likely to buy highbrow wines on the retailers webshop in 2017.

Solution:

- Based on 2016 available data, build a Look-alike model which predicts the most likeliness to target the highbrow wine customers in 2017.

- Look-alike modeling *"results in double or even triple the results of standard targeting, according to the 30 percent of advertisers and more than half of agencies who reported using the tactic."*

- Using Data science, we can treat this as a <u>Classification problem</u>

Context & Briefing - Points

- **Goal:** Develop a Look-alike model for Highbrow wine online customers and predict the results for 2017 based on this model

- **Relevant information:** 2016 labeled data of the customers, 2017 unlabeled data of the customers

- **Deliverables:**
  - 2017 predictions for Highbrow wine customers and code block used for development

- **Validation metrics used:**
  - Prediction metrics on validation datasets (Confusion matrix, ROC curve)
  - Prediction vs Actuals for the unlabeled data (*Handled by Validators*)

Context & Briefing - Modeling approach

Data discovey

- Data collection
- Data Exploration

Data Preparation

- Data transformation
- Encoding & Scaling

Modeling Pipeline

- Feature Selection
- Model selection
- Hypertuning

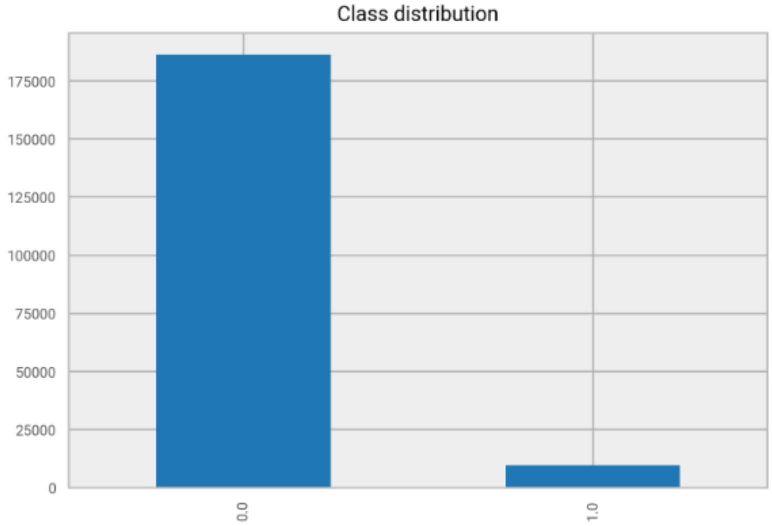
Train & Score

- Predict the labels
- Model Evaluation
- Scoring the Unlabeled data

Data Discovery – Imbalanced datasets

- Target Class distribution is highly imbalanced

```
target variable distribution:
class 0.0 : 186200
class 1.0 : 9484
Class distribution ratio: 19.63 : 1
```

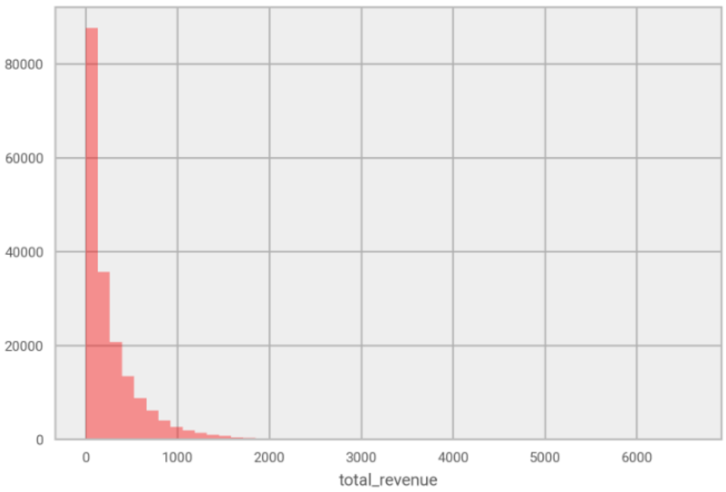


- Solution for Modeling:
  - Use Re-sampling techniques to better predict the minority class during training.
  - Use class weight balancing and stratified sampling during training.

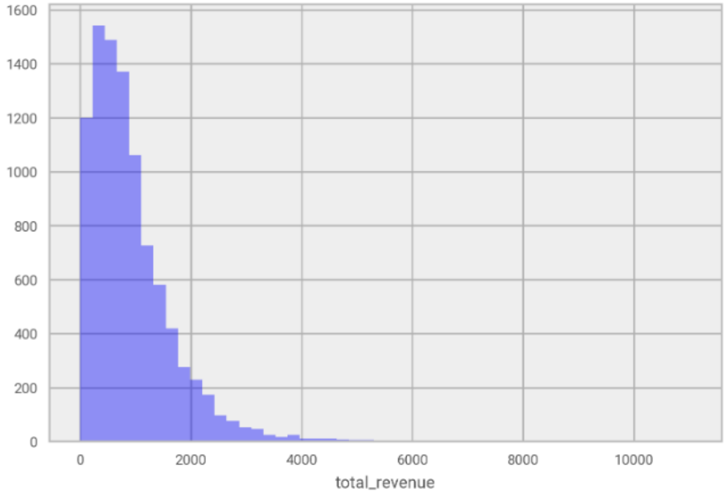
Data Discovery – Revenue distribution

- Feature “total_revenue”
- Class 0 shows many customers as less active, but Class 1 has proper distribution of total revenue

Class 0



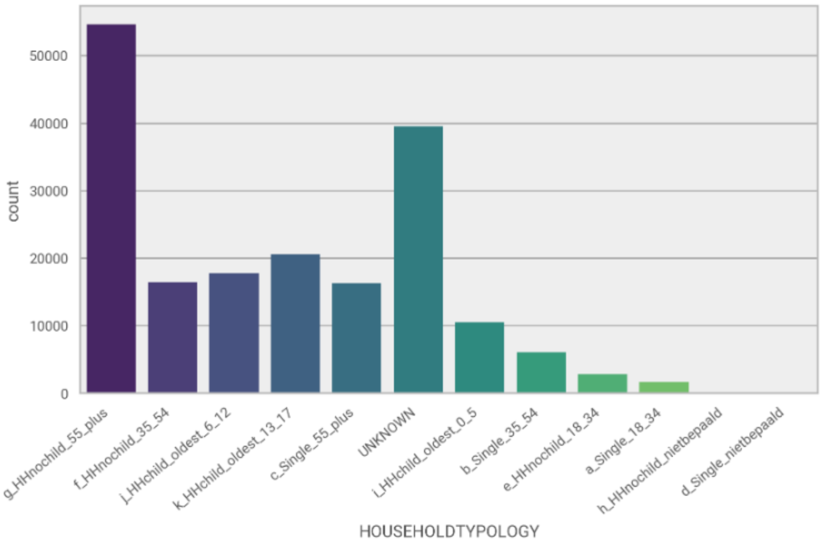
Class 1



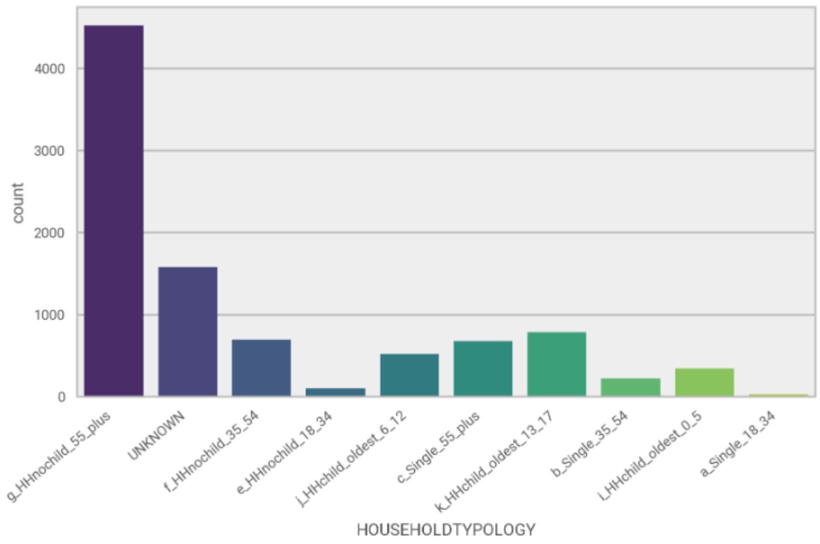
Data Discovery – Household typology

- Feature “HOUSEHOLDTYPOLOY”
- ‘HHnochild_55_plus’ customers stands out for both Class 0 and 1. This could be due to loyal customers as pensioners and having same behavior online and offline

Class 0

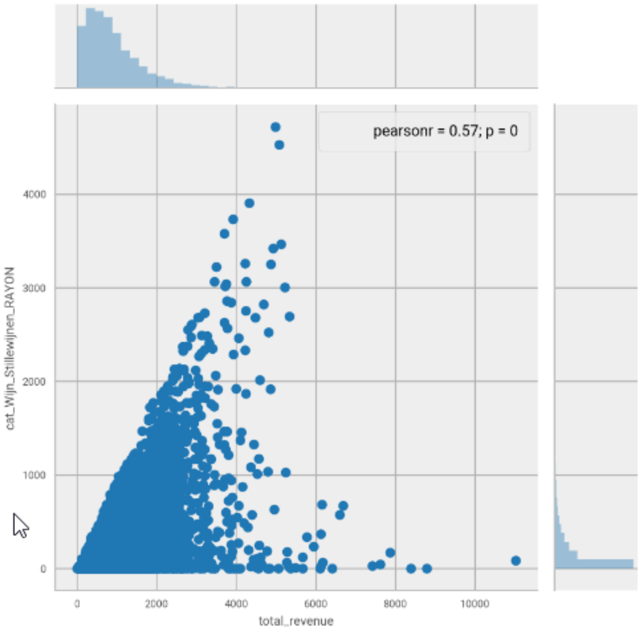


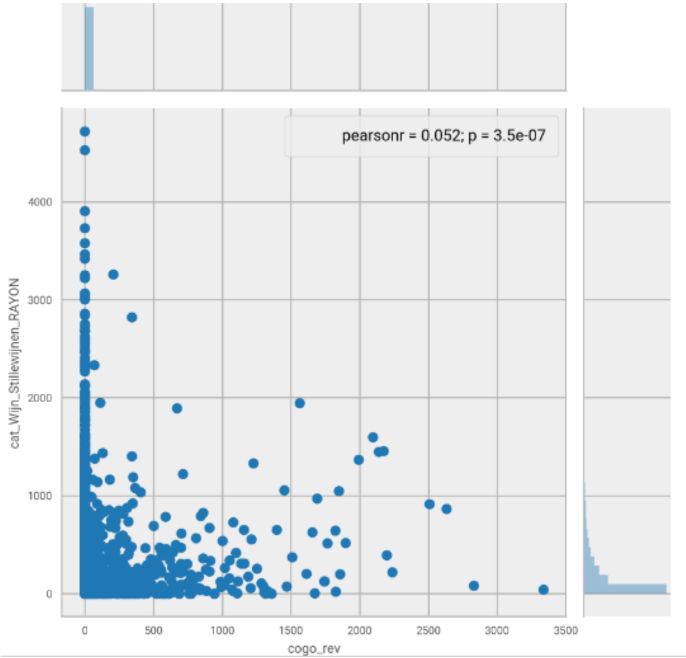
Class 1



Data Discovery – Wijn & Revenue relation

- Finding relation between “Wijn” turnover with “total revenue” and “collect&go revenue”
- Wijn and total revenue are correlated, but no explicit behavior between collect&go revenue





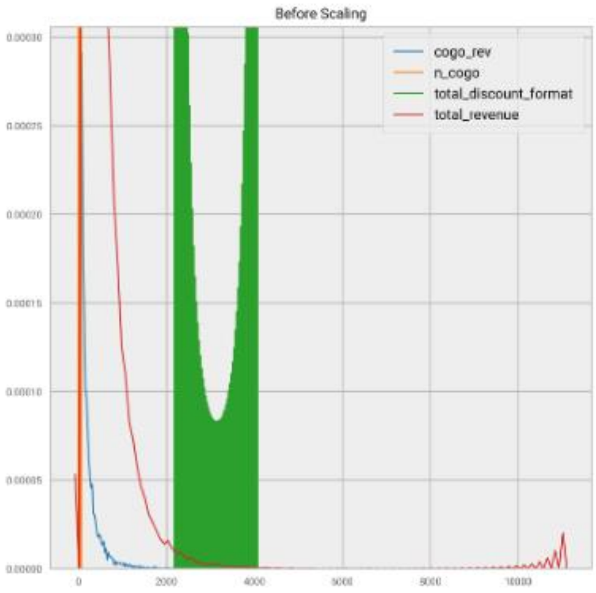
Data Preparation - Transformation

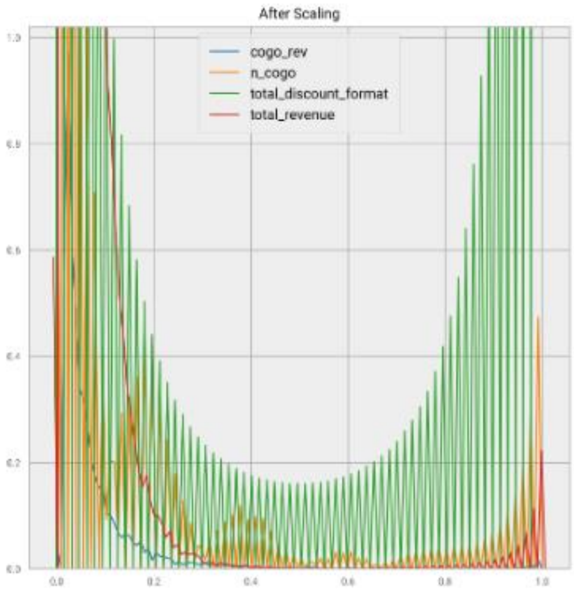
Below data transformation steps are performed to clean the data;

- Removed target variables with null values (can’t be trained)
- Removed ‘Jaar’ column which was constant across the data
- Enriched missing/null/negative values
  - Collishop_customer (‘Null’ to ‘N’)
  - Total_Revenue (‘Null’ to mean value)
  - Category turnovers (‘Null’ and negative to 0)
- Transform Negative to Positive
  - Price sensitivity and Total discount values are stored as negative values
  - Absolute values are considered for transformation
- Removing Outlier observations (based on frequency and turnover in SOW_type_colr)

Data Preparation - Encoding

- Used OnHotEncoding for categorical variables
- Used MinMaxScaler to scale the features between 0 and 1





Modeling Approach – Feature Selection

The following methods were used for feature selection and used voting to select the best predictors from them;

- Information Value using WOE
- FE using Random Forest
- Chi Square best variables
- RFE using Logit

In total **23 best features** were selected

(out of 55 features)

Features	IV	RF	Chi_Square	FE	final_score
SOW_colr	1	1	0	1	3
total_discount_format	1	1	0	1	3
cat_Wijn_Stillewijnen_RAYON	0	1	0	1	2
rev_ticket	0	1	0	1	2
cat_AP_STDR_WhiskyONLINE	0	0	1	1	2
Collishop_customer_Y	0	0	1	0	1
HOUSEHOLDTYPOLOGY_g_HHnochild_55_plus	0	0	1	0	1
SOW_type_colr_UNKNOWN	0	0	1	0	1
cat_Babyluiers	0	0	0	1	1
cat_Ber_Ger_VersMaaltijdsalades	0	0	0	1	1
cat_Bier_Genietbieren	0	0	0	1	1
cat_BroodKorthoudbaar	0	0	0	1	1
cat_Chips	0	0	0	1	1

Modeling Approach – Model Selection

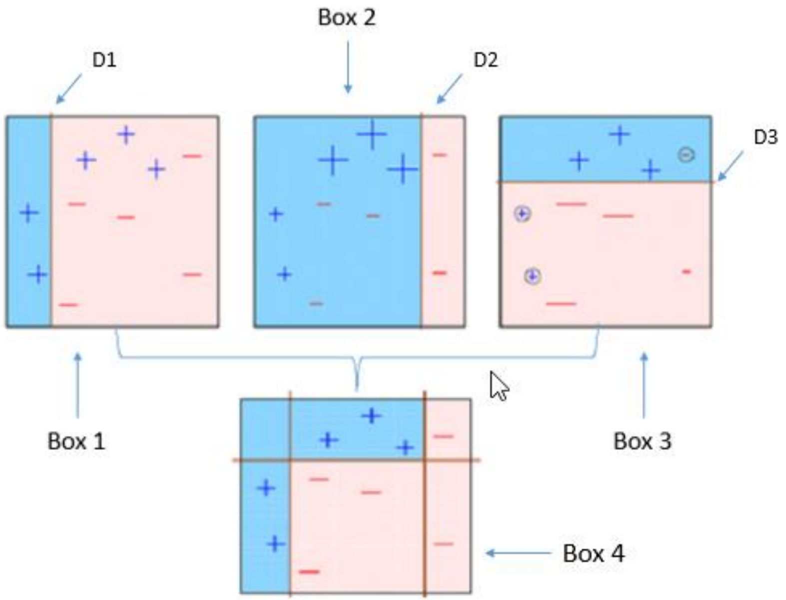
	Random Forest	XGBoost	Stacking (multiple algorithms)
Description	Tree ensemble model, uses bagging to make predictions	Tree ensemble model, uses gradient boosting to make predictions	Ensemble model of different algorithms and final meta-model does predictions from the results of level1 estimator models
Pros	Easy to implement, options to fine-tune parameters, shows feature importances	Easy to implement, options to fine-tune parameters, model adjusts the training errors	Gets good predictions because all models learn something new, which improves the final predictive power
Cons	Not a great predictor with low observations, Easily overfits with less number of train dataset.	Explanation of feature importance is difficult and not completely reliable because of boosting technique	Model execution takes longer time to run, Not intuitive in validating the results

XGBoost and Stacking models were used for this project;

- Both models gave interesting results
- Usage of "Model Business Value Framework" can be used to decide the final results

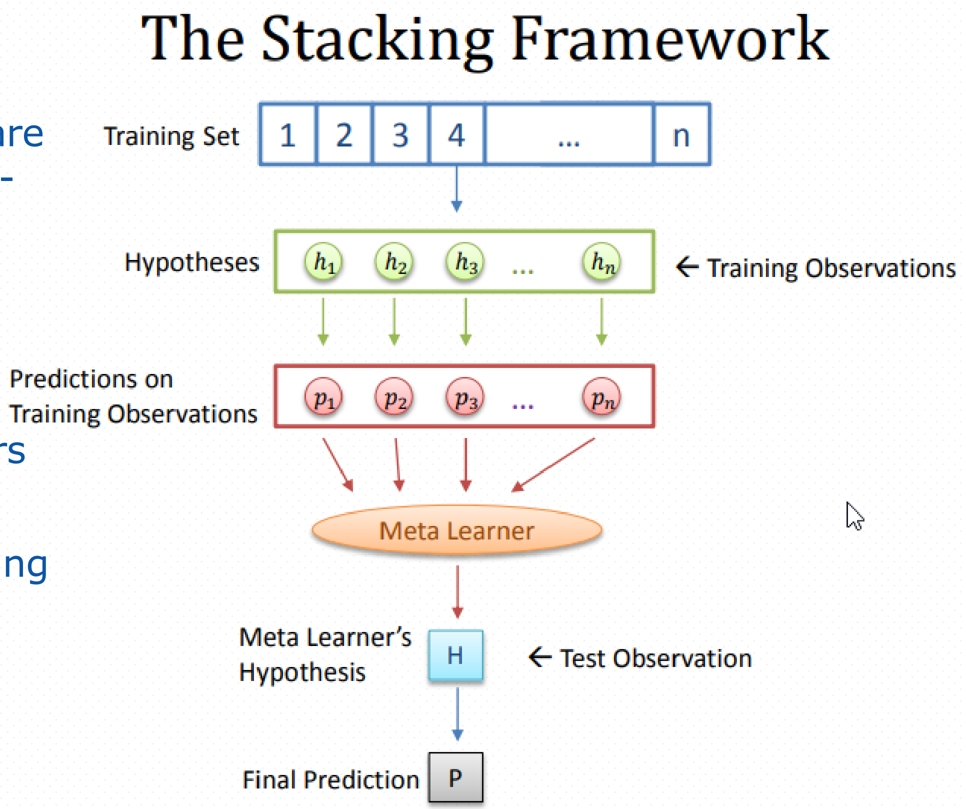
Modeling Approach – XGBoost Pipeline

- Optimized distributed gradient boosting algorithm, known for its speed and performance.
- Core algorithm is parallelizable
- Consistently outperforms other algorithm methods
- Wide variety of tuning parameters



Modeling Approach – Stacking Pipeline

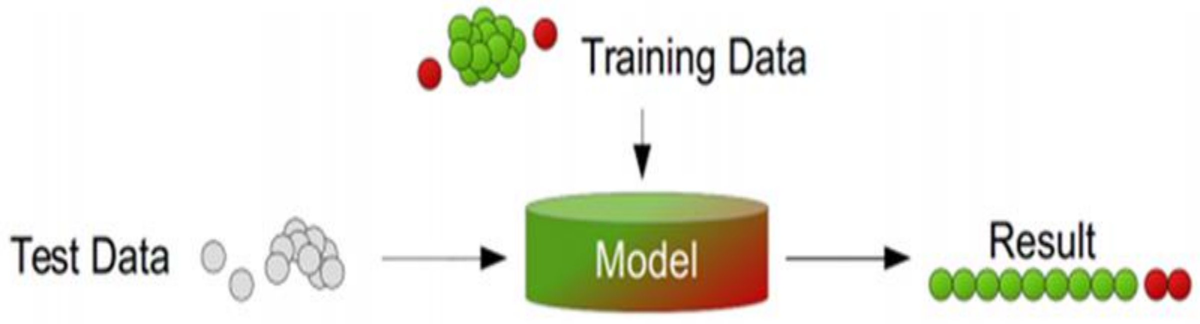
- Stacks multiple estimator models, which are normally different learning path and meta-model aggregates and learns from their results to get powerful predictions
- Mostly used in Kaggle competitions to get better output
- Can add n number of estimators and layers for intensive training the model
- Takes more time for execution when training set is large



Modeling Approach

Training the model:

- Prepare the ML pipeline with different options (XGBoost, Random Forest, Stacking)
- GridSearch technique is used to tune the hyper parameters
- Re-sampling option is used for one of the iteration to compare the results.
- Model is evaluated for the Classification metrics
- Trained model and validation results are saved in a pickle file.



Modeling Approach – XGBoost

Training the model:

- Over-sampling the minority class – SMOTE algorithm
- Hyper tuned parameters

n_estimators= 150,
colsample_bytree=0.8,
gamma=0.5,
learning_rate=0.7,
max_depth=5,
min_child_weight=1.5,
reg_alpha=0.75,
reg_lambda=0.45,
nthread=6,
scale_pos_weight=0.5,
subsample=0.9

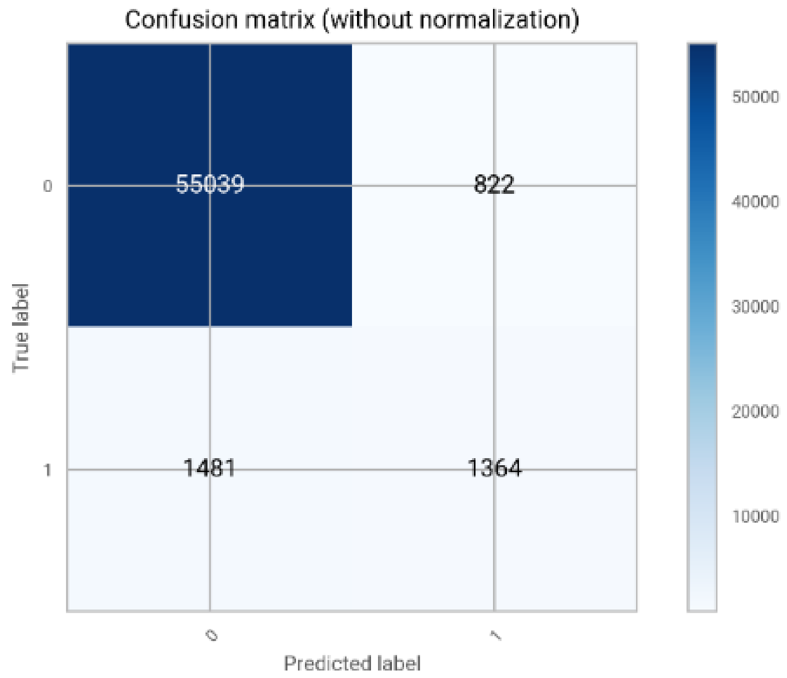
Modeling Approach – XGBoost

Model evaluation:

- Trained model is validated with test data predictions and evaluated with below metrics
- Metrics used -> Model accuracy , Confusion matrix, Precision, Recall and ROC curve

Accuracy = **96,1%**

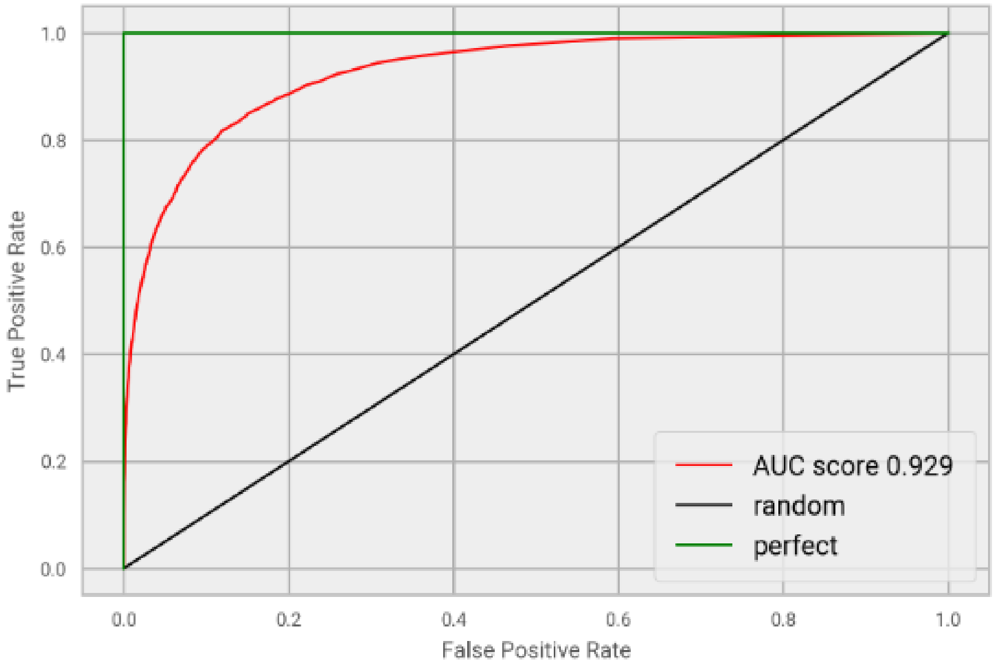
```
Model accuracy: 0.961
Classification report:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98     55861
           1       0.62      0.48      0.54      2845
```



Modeling Approach – XGBoost

Model evaluation:

- Even though model accuracy is very good, real goal of this model is to predict Class 1 customers (which is not that great)
- As we see, for Class 1, Recall = **48%** & Precision = **62%**
- AUC score and ROC curve shows this is the maximum model could achieve.
- AUC score = **92%**



Modeling Approach – XGBoost

Model scoring:

- The trained and evaluated model is used to predict the un-labeled data of 2017 (scoring)
- After scoring, the new predictions along with prediction probabilities are saved in CSV file

After Scoring:		
Total Customers scored	**190.740**	Percentage targeted
Customers targeted	**17.931**	**9%**
Expected value of model	**€ 20.163**	

masked_customer_id	highbrow_wines_prediction	prediction_probal
339834	0	0,047
339837	0	-
339848	1	0,941
339864	1	0,511
339866	1	0,995
340067	1	0,885
340189	1	0,966
340281	1	0,862

Modeling Approach – Stacking

Training the model:

- Over-sampling the minority class – SMOTE algorithm

- L1 estimator models used => Random Forest Classifier, AdaBoost Classifier, XGB Classifier, Support Vector Machine Classifier

- Meta-learner model used => Logistic Regression

- Hyper-parameters => Variant A (for out-of-fold train datasets), ROC AUC metric score (for tuning auc metrics), Stratified boolean (to handle the imbalanced data)

Modeling Approach – Stacking

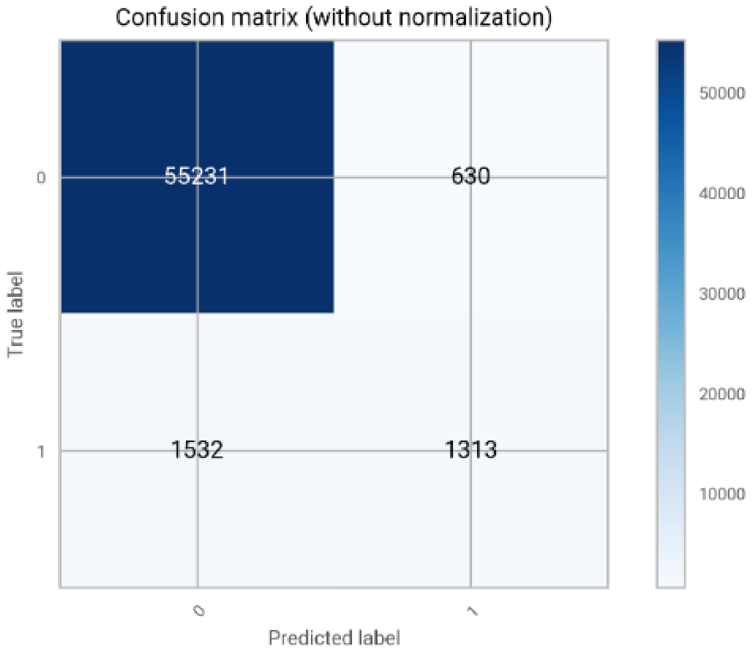
Model evaluation:

- Trained model is validated with test data predictions and evaluated with below metrics
- Metrics used -> Model accuracy , Confusion matrix, Precision, Recall and ROC curve

Accuracy = **96,3%**

Model accuracy: 0.963
Classification report:

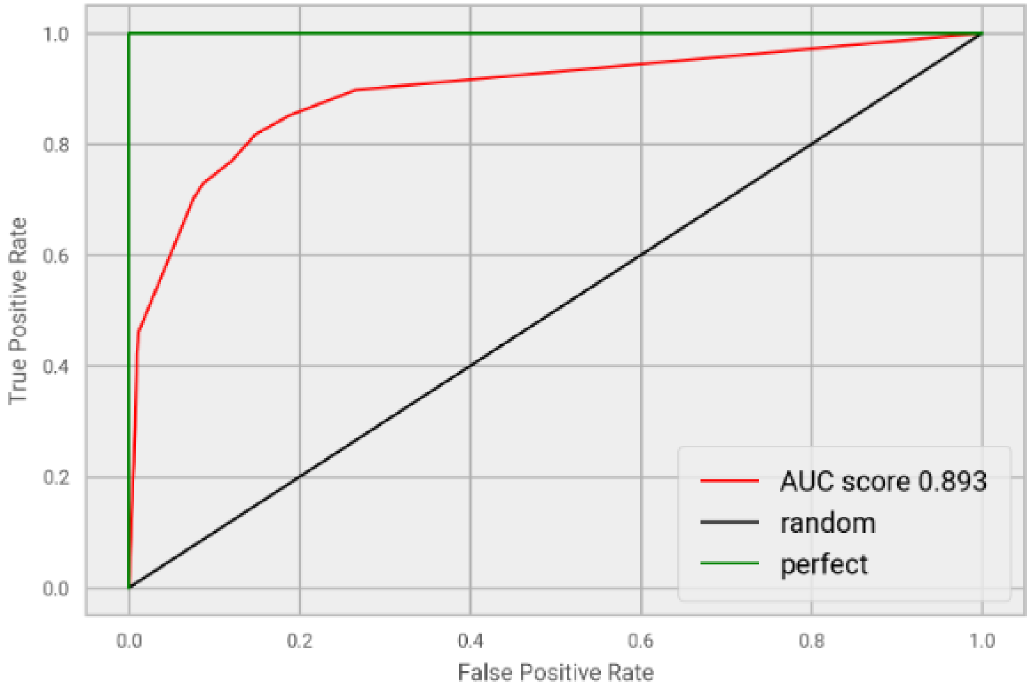
	precision	recall	f1-score	support
0	0.97	0.99	0.98	55861
1	0.68	0.46	0.55	2845



Modeling Approach – Stacking

Model evaluation:

- Even though model accuracy is very good, real goal of this model to predict Class 1 customers is not that good.
- As we see, for Class 1, Recall = **46%** & Precision = **68%**
- AUC score and ROC curve shows this is the maximum model could achieve.
- AUC score = **89%**



Modeling Approach – Stacking

Model scoring:

- The trained and evaluated model is used to predict the un-labeled data of 2017 (scoring)
- After scoring, the new predictions along with prediction probabilities are saved in CSV file

After Scoring:			
Total Customers scored	190.740	Percentage targeted	
Customers targeted	47.981	25%	
Expercted value of model	€ 52.068		

masked_customer_id	highbrow_wines_prediction	prediction_probability
339793	0	0,019
339806	1	0,952
339807	0	0,019
339809	0	0,057
339812	0	0,019
339815	0	0,019

Decision Making – Comparing models

Comparing Models thru values:

- Accuracy score can be misleading + Evaluation metrics of models are almost same here.
- Best model depends on *what* we want to achieve with the model
- Using a common framework with "Model Business Value" calculation can be used to compare and decide on the model to use

XGBoost model:

Accuracy = 96,1%

F1 score = 54%

Customers targeted =9%

Stacking Model

Accuracy = 96,3%

F1 score = 55%

Customers targeted =25%

Decision Making – Model Business Value

Using the values of Confusion Matrix, Cost/Benefit for the business can be computed (Expected Value of a model)

$$\boldsymbol{EPMi} = P(Y,p).B(Y,p) + P(Y,n).B(Y,n) + P(N,p).B(N,p) + P(N,n).B(N,n)$$

Where,

P(Y,p) = probability of predicted class = Yes and true class (Positive (p))

B(N,n) = benefit when predicted class = No and true class (Negative (n))

Model Business Value

$$\begin{aligned}\boldsymbol{EP}(\boldsymbol{Mi}) &= \boxed{P(Y \setminus p)}.\boxed{P(p)}.B(Y,p) + \boxed{P(Y \setminus n)}.\boxed{P(n)}.B(Y,n) \\ &+ \boxed{P(N \setminus p)}.\boxed{P(p)}.B(N,p) + \boxed{P(N \setminus n)}.\boxed{P(n)}.B(N,n)\end{aligned}$$

Computed from Confusion matrix

$$\begin{aligned}\boldsymbol{EP}(\boldsymbol{Mi}) = {} & P(p).[Recall.B(Y,p) + FPR.B(N,p)] + \\ & P(n).[FNR.B(Y,n) + Specificity.B(N,n)]\end{aligned}$$

Decision Making – XGBoost Expected Value

Cost of campaign	€	**1**	per customer
Margin of sales	€	**50**	per customer

True Valu		Predicted Values		
	Predictions	Negative (0)	Positive(1)	
	Negative (0)	55039	822	55861
	Positive(1)	1481	1364	2845
		56520	2186	**58706**

True Valu		Predicted Values		
	Probability	Negative (0)	Positive(1)	**P(true class)**
	Negative (0)	98,5%	1,5%	95,2%
	Positive(1)	52,1%	47,9%	4,8%

True Valu		Predicted Values	
	Benefits	Negative (0)	Positive(1)
	Negative (0)	0	-1
	Positive(1)	0	49

Computing the Expected Value

True Valu						
	Benefits	E(N/n)	E (N/p)	E(./n)	En = P(n).E(./n)	E = Ep+En
	Negative (0)	0	-0,01	-0,01	**-0,0140**	€ **1,12**
	Positive(1)	0	23,49	23,49	**1,1385**	
		E(N/p)	E (Y/p)	E(./p)	Ep = P(p).E(./p)	

After Scoring:		
Total Customers scored	**190.740**	Percentage targeted
Customers targeted	**17.931**	**9%**
Expected value of model	**€ 20.163**	

Decision Making – Stacking Expected Value

Cost of campaign	€	**1**	per customer
Margin of sales	€	**50**	per customer

True Values		Predicted Values		
	Holdout	Negative (0)	Positive(1)	
	Negative (0)	55231	630	55861
	Positive(1)	1532	1313	2845
		56763	1943	**58706**

True Values		Predicted Values		
	Benefits	Negative (0)	Positive(1)	
	Negative (0)	0	-1	
	Positive(1)	0	49	

True Values		Predicted Values		
	Probability	Negative (0)	Positive(1)	**P(true class)**
	Negative (0)	98,9%	1,1%	95,2%
	Positive(1)	53,8%	46,2%	4,8%

Computing the Expected Value

True Values	Benefits	E(N/n)	E (N/p)	E(./n)	En = P(n).E(./n)	E = Ep+En
	Negative (0)	0	-0,01	-0,01	**-0,011**	€ **1,09**
	Positive(1)	0	22,61	22,61	**1,096**	
		E(N/p)	E (Y/p)	E(./p)	Ep = P(p).E(./p)	

After Scoring:			
Total Customers scored	**190.740**	Percentage targeted	
Customers targeted	**47.981**	**25%**	
Expercted value of model	**€ 52.068**		

Decision Making – Final

Which Model and Results to be used ?

With the input on Cost – Benefits details of the campaign, we will be able to decide which results could be finalized (to be discussed in tomorrow's meeting) !

Deliverables

- **Lookalike_model_results_2017.xlsx** => CustomerID, Highbrow wines predictions (0 or 1), prediction probability
  - 3 result sheets are given (Model1 and Model2 results to be considered)

- **ValueBasedFramework.xlsx** => To be discussed during the meeting tomorrow

- **code_03_Lookalike_proj** => python codes following the functional programming structure (Modules and Functions)

  - Source codes are python (.py) files
  - Code is developed in modularized approach
  - Common functions and ML algorithms are encapsulated in to separate modules and functions
  - Execute 00_main.py source to execute the full flow

Deliverables – Code flow

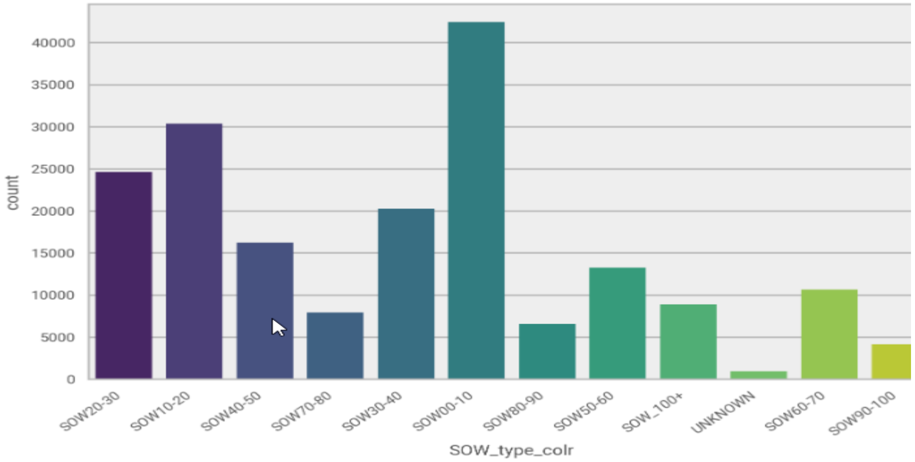
- _03_Lookalike_proj
  - input_files
  - modules
    - __pycache__
    - __init__.py
    - fn_data_io.py
    - fn_featselect_weightofevid.py
    - fn_gridsearch_validate.py
    - mod_01_datapreparation.py
    - mod_02_dataexploration.py
    - mod_03_featureselection.py
    - mod_04_modeltraining.py
    - mod_05_modelscoring.py
  - output_files
  - requirement
  - testing
  - 00_main.py
  - README.md

```python
# Author              : Arif Thayal
# Project name        : _03_Lookalike_Model
# Purpose             : Main python code to execute
# Last modified by    : Arif Thayal
# Last modified date  : 09/05/2019
# --------------------------------------------------

# import the libraries
import os, sys, importlib
import pandas as pd
import seaborn as sns
import numpy as np
from matplotlib import pyplot as plt
from datetime import datetime, timedelta, date

%matplotlib inline
pd.options.display.html.table_schema = True

project = '_03_Lookalike_proj'
sys.path.append('./'+project+'/modules/')

# define the directory variables
input_dir = os.path.join(project,'input_files')
output_dir = os.path.join(project,'output_files')
code_dir = os.path.join(project,'src')

```

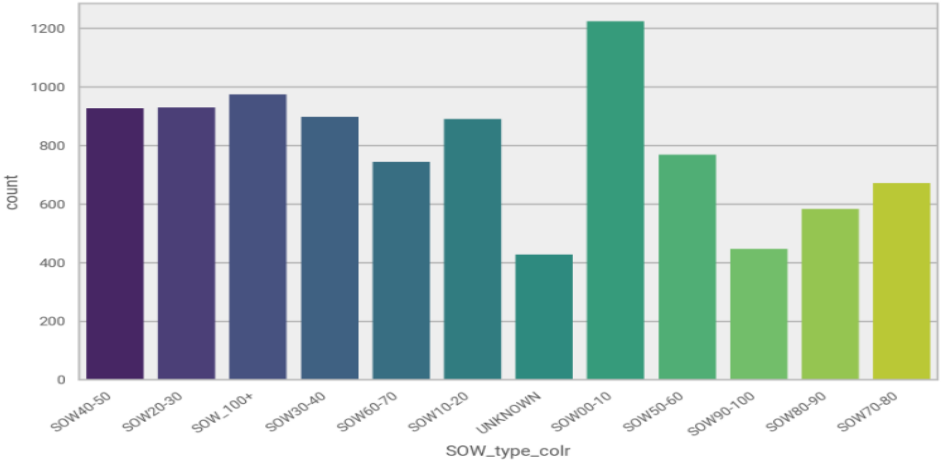
Appendix

Data Discovery

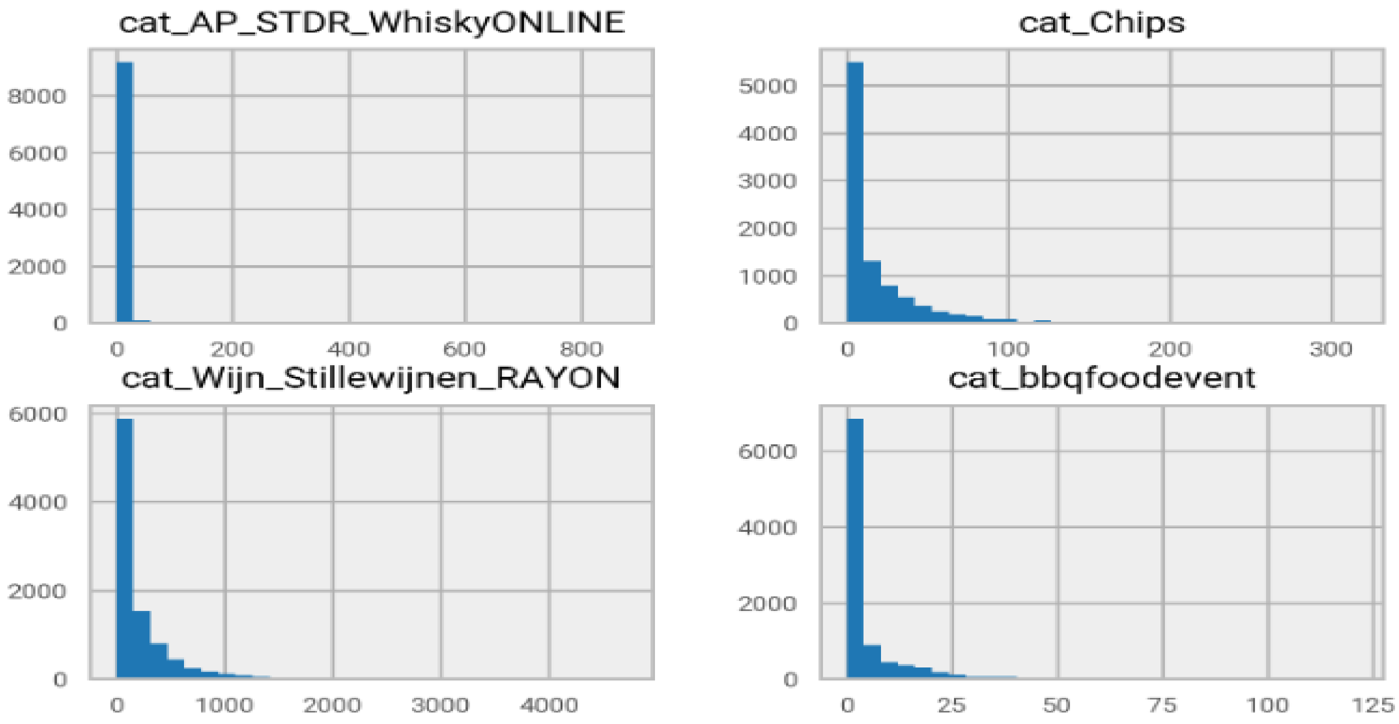
Class 0



Class 1



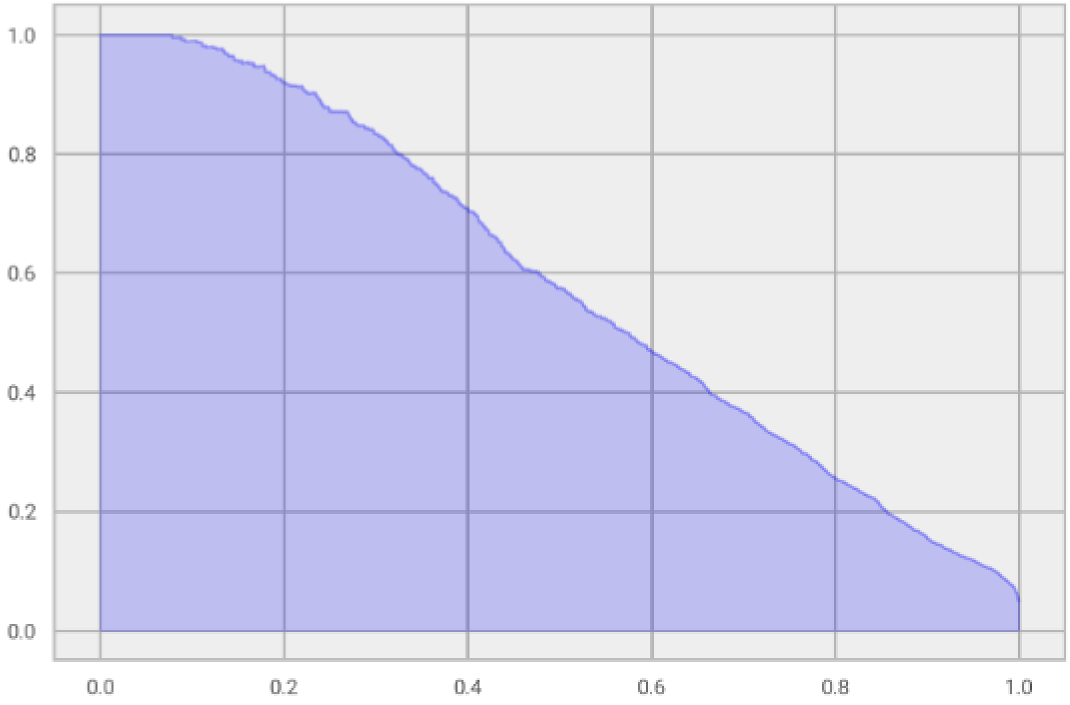
Data Discovery



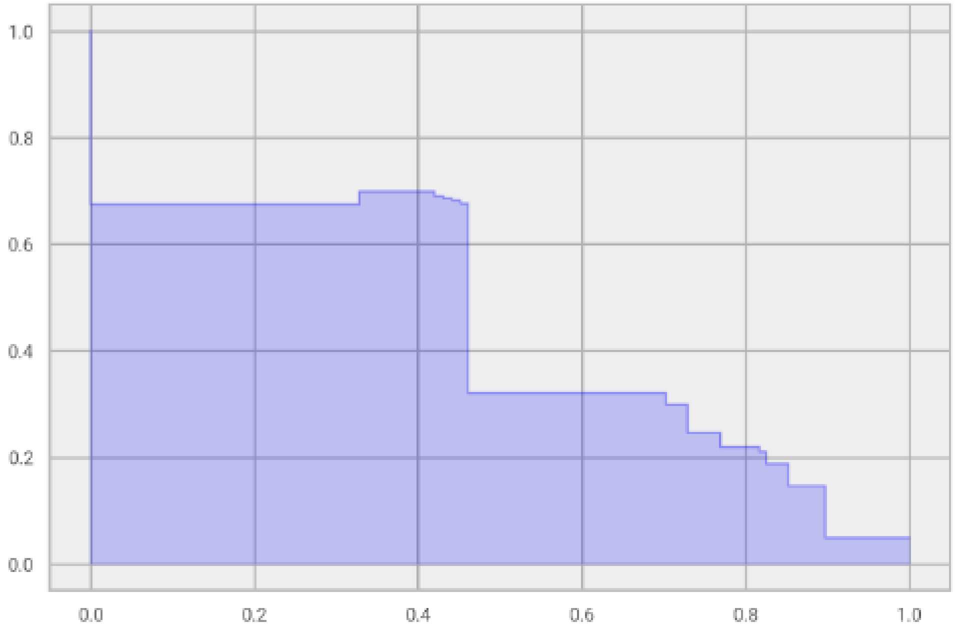
Features Selected

Features	IV	RF	Chi_Square	FE	final_score
total_revenue	1	1	1	1	**4**
SOW_colr	1	1	0	1	**3**
total_discount_format	1	1	0	1	**3**
cat_Wijn_Stillewijnen_RAYON	0	1	0	1	**2**
rev_ticket	0	1	0	1	**2**
cat_AP_STDR_WhiskyONLINE	0	0	1	1	**2**
Collishop_customer_Y	0	0	1	0	**1**
HOUSEHOLDTYPOLOGY_g_HHnochild_55_plus	0	0	1	0	**1**
SOW_type_colr_UNKNOWN	0	0	1	0	**1**
cat_Babyluiers	0	0	0	1	**1**
cat_Ber_Ger_VersMaaltijdsalades	0	0	0	1	**1**
cat_Bier_Genietbieren	0	0	0	1	**1**
cat_BroodKorthoudbaar	0	0	0	1	**1**
cat_Chips	0	0	0	1	**1**
cat_Houtpelletskolen_briketten	0	0	0	1	1
cat_Incontinentie_luiers	0	0	0	1	1
cat_KaasSeizoenskazen	0	0	0	1	1
cat_Kauwgum	0	0	0	1	1
cat_KoudeSauzen	0	0	0	1	1
cat_Notengedroogdfruit_groenten	0	0	0	1	1
cat_Ontbijtgranen_Volwassenen	0	0	0	1	1
cat_VNCBerBurgers	0	0	0	1	1
cat_VerseKaasFruitkazen	0	0	0	1	1
n_tickets	1	0	0	0	1
price_sens_colr_format	1	0	0	0	1

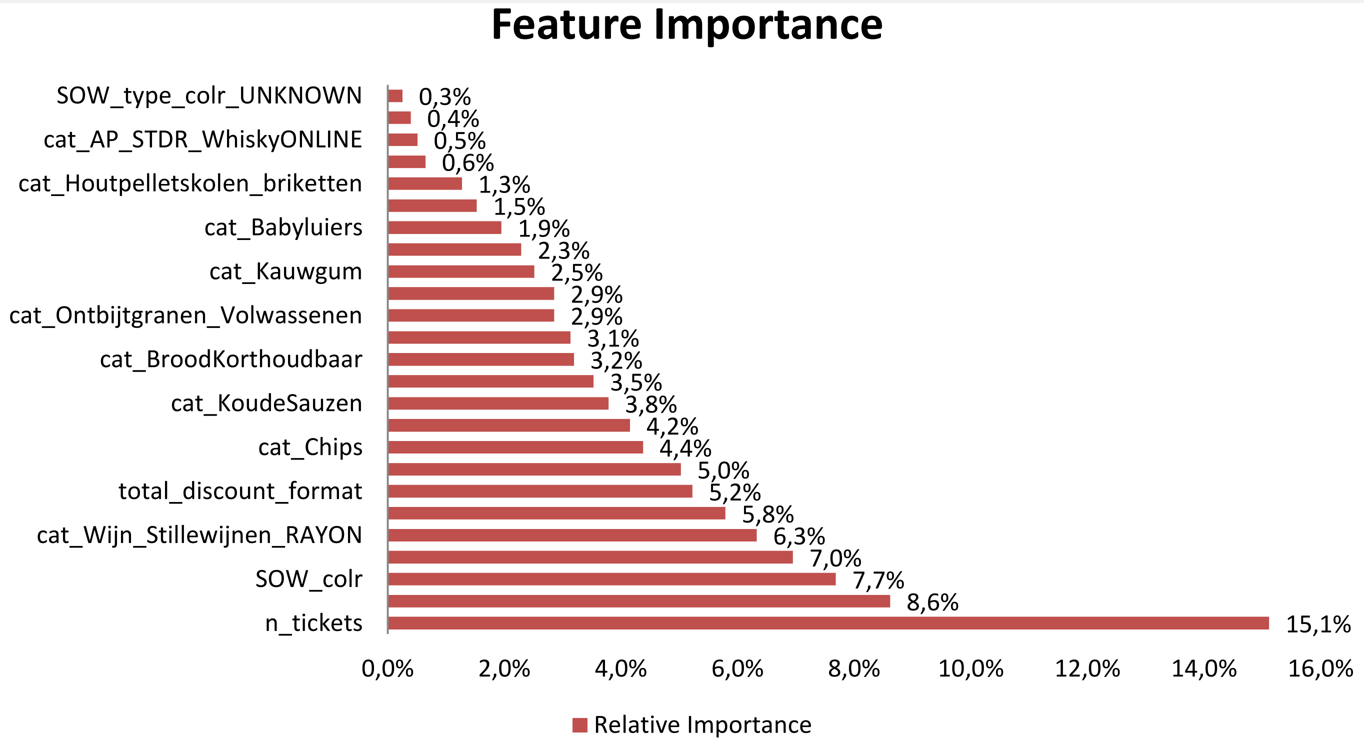
Precision Recall Curve - XGBoost



Precision Recall Curve - Stacking



Model Feature Importance



Questions?

Technology Trends – Retail Industry

"Disruptive change has come to the supermarket sector. Technological innovations and analytics usage for online and in-store, as well as shifting consumer expectations, are changing the way food retailers operate."

Below 5 upcoming trends to focus;

1. Tech-transformation in E-commerce
2. Digital transformation of physical stores
3. Advancing "social commerce"
4. Tech advancement in supply chain
5. Data and information traceability

Technology Trends – E-commerce

“E-commerce have been a disruptive force, taking market share from traditional bricks-and-mortar retailers”

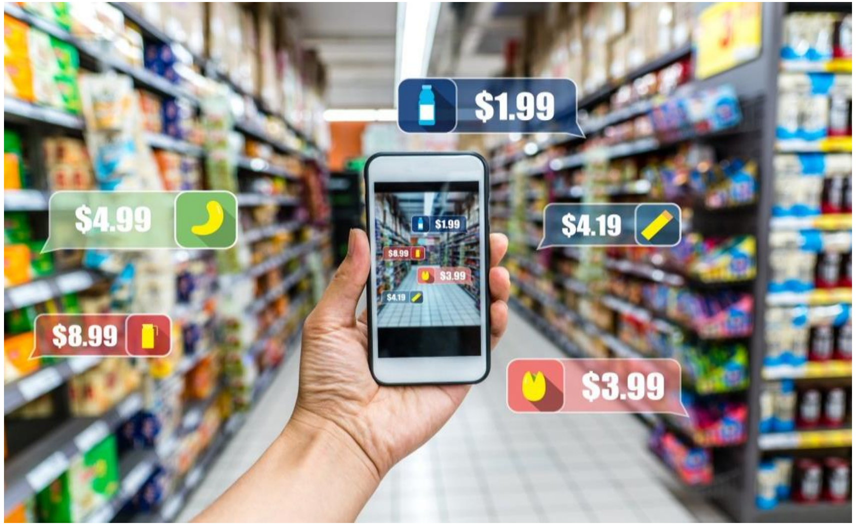
- Omni-channel -> Improving the shopper experiences combining online and offline needs
- Requirement of fresh-food online
- Warehouse automation and AI for logistics operations and personalization per region

Technology Trends – Digital Stores

"Technological advances are changing the course how people shop for groceries in-store"

- Physical stores offer more digital experiences
  - Guiding shoppers thru in-store aisle
  - Shopping-cart mounted devices
  - Sensors connecting POS and carts

- Personalized recommendations for in-store consumers



Technology Trends – Social Commerce

"The evolution of e-commerce could result in new ways of shopping – more social and instantaneous"

- Retailers making every moment shoppable
- Social networks deliver targeted marketing, with instant buy options
- Online photos, videos, ads makes it more convenient and simpler to shop

Technology Trends – Supply chain tech

"Supply chain needs to know in precise what is coming-in from field, what is in-storage and what is the demand to be more efficient"

- Many production units use internet-of-things. These data can be utilized to know the incoming stock
- On-shelf availability programs can be used to know the existing storage
- These advancements could significantly reduce food waste, supermarket's floor space and storage requirements

Technology Trends – Traceability

"Gaining consumer confidence with access to detailed information on the origin of products"

- Block chain technology is used for getting the complete lifecycle of products
- Improved access to data will extend to nutrition and taste
- It will also be used for personalized recommendation of recipes and food pairings