Predictive Analytics on Operation Thunderstorm

Indicating which customers are most likely to make a purchase in the near future.

Samenvatting

In deze rapport wordt een model gemaakt waarmee je zou kunnen voorspellen of in de toekomst een klant terugkomt. Het doel van onderzoek is om management te kunnen overtuigen dat ze voor toekomstige marketingplannen beslissingen nemen gebaseerd op data. Er wordt op basis van verkoopdata acht features gemaakt die informatie over klant geeft. Dan wordt er gekeken of er correlatie tussen de features zitten en welke feature het beste voorspelling geeft. Er wordt dan door middel van algoritmes op data getraind om voorspellingen te kunnen doen. Door gebruik te maken van data kan je beslissingen nemen voor welke strategie je dan moet kiezen om meer omzet te kunnen boeken.

15-04-2019 Kousar Sedigi & Twumi Weterings

Inleiding

Makers van Thunderstorm willen graag advies over hoe ze meer omzet kunnen krijgen op basis van hun verkoopdata. Daarvoor willen ze gebruik maken van data algoritmes om te kijken welke klanten in de toekomst nog een aankopen zullen gaan doen. Op deze manier willen ze weten welke klanten loyaal zijn en daarop hun marketingstrategie aanpassen. Daarvoor zullen we gebruik maken van data mining technieken op de verkoopdata.

Doel

Het doel van het rapport is om het management te overtuigen dat ze klanten beter kunnen begrijpen door data te gebruiken die zij beschikken. Hiervoor gaan we een pilot draaien op hun verkoopdata en laten we zien dat gedrag van klanten beter te analyseren is. Daarom luidt ons hypothese:

Gedrag van klanten kan geanalyseerd worden door te gebruik maken van datatechnieken om beslissing te nemen om omzet te kunnen vergroten.

Daarom gaan we in dit rapport het gedrag van de klanten voorspellen door verschillende data algoritmes toe te passen. We hebben tot onze beschikking de data van verkopen.
Voordat we beginnen splitsen we de data op in 2 delen. Eerste deel bevat gegevens over klanten die tot 2014-01-01 lopen en tweede deel bevat gegevens over klanten vanaf 2014-01-01

Methodologie

Voordat we voorspellingen kunnen doen om klant gedrag te kunnen analyseren, maken we een nieuwe tabel waar we de verkoopdata omzetten naar klanten data. Eerst splitsen we de data in twee delen door volgende functie.

```
sales = sales.sort_values(["saleDateTime"])
split1 = sales['saleDateTime'] < '2014-01-01 00:00:00'
sales14 = sales[split1]
split2 = sales['saleDateTime'] > '2013-12-31 23:59:59'
sales15 = sales[split2]
```

Dan leiden 8 *features* af van de verkoopdata van eerste deel die ons kunnen helpen om klant gedrag te kunnen meten. Sommige features zijn op basis van literatuur afgeleid en andere juist door logica te gebruiken. En tweede gedeelte kijken we of onze voorspelling wel over een komen met de tweede gedeelte van de data.

De *features* gaan ons helpen om te kijken of een klant blijft terugkomen. Acht *features* die wij hebben afgeleid zijn:

1. We gaan ervan uit dat iedereen op vakantie gaat. Dus als je uit verschillende landen het spel speelt, dan ben je een klant die gehecht is aan het spel, a 'loyal' customer. De vraag is dan: Blijft de klant terugkomen?
2. Hoe langer iemand het spel speelt, groter de kans dat hij blijft komen. (Takes, 2019)
3. Doet iemand die veel geld uitgeeft ook eerder een nieuwe aankoop?
4. Doet iemand die al veel aankopen heeft gedaan in het verleden ook sneller een nieuwe aankoop? (2003).
5. Is er een verband tussen de betaalmethode en hoe snel iemand nog een aankoop doet?
6. Geografische locatie. Mensen die een bepaalde landen wonen, blijven terugkomen (2009, August).
7. Transacties. Blijven mensen die steeds meer aankopen doen ook terugkomen (2013).
8. Besteding. Blijven mensen die een groter bedrag uitbesteden doen ook terugkomen.

Een korte toelichting hoe deze features afgeleid zijn van de functie.

1. **customers['countries']= sales14.groupby("accountName")["ipCountry"].nunique()**
   Voor de eerste feature hebben wij alles gesorteerd op accountName en daar gekeken naar hoeveel landen iemand is geweest.
2. **customers['lifespan']= sales14.groupby("accountName")["saleDateTime"].max() - sales14.groupby("accountName")["saleDateTime"].min()**
   Hier pakken we de tijd van laatste verkopen en eerste inkoop en tellen hoeveel tijd er tussen zit.
   customers['lifespan'] = customers["lifespan"].dt.total_seconds()
   Hier zorgen we dat het in aantal sec wordt weergegeven gezien er soms maar seconden verschil zit.
3. **customers['lifetime_spend']= sales14.groupby("accountName")["priceInEUR"].sum()**
   Hier pakken we het totale bedrag per klant.
4. **customers['lifetime_trans']= sales14.groupby("accountName")["saleId"].nunique()**
   Hier pakken we het aantal transacties per klant.
5. **tempx = sales14.groupby("accountName")["ipCountry"].value_counts()**
   Hier gaan we eerst aantal keer het land optellen waar de klant is geweest
   temp5 = tempx.groupby("accountName").idxmax()
   Daarna pakken we per klant het land waaruit hij het meest heeft gekocht. (waar hij woont nemen we dan aan)
   sales["ipCountry"].value_counts().head(5)
   fav_countries = ["FR","DE","PL","GP","NL"]
   Woon je in de top 5 landen dan is de kans groter dat je terug komt
6. **tempy = sales14.groupby("accountName")["methodId"].value_counts()**
   temp6 = tempy.groupby("accountName").idxmax()
   Dit pakken we op hetzelfde manier aan als bij 5.
   sales["methodId"].value_counts().head(3)
   fav_paymethods = [2000,1000,40]
   Heb je method id 2000, 1000 of 40 als favoriete betaalmethode dan blij je terugkomen.
7. **first = sales14.groupby("accountName")["saleDateTime"].nsmallest(2)**

Omdat de stijging moeilijk te meten was, hebben we gekozen om het verschil tussen twee kleinste aankopen te meten en twee grootste.
Hier zoeken naar de datum van de twee eerste aankopen
last = sales14.groupby("accountName")["saleDateTime"].nlargest(2)
Hier zoeken we naar de datum van de twee laatste.
abs(first.diff().reset_index().groupby("accountName").tail(1).set_index("accountName")["saleDateTime"].dt.days)
df['Last']=
abs(last.diff().reset_index().groupby("accountName").tail(1).set_index("accountName")["saleDateTime"].dt.days)
Hier kijken we of het verschil groter is of kleiner. Is het kleiner dan betekent het dat je veel meer inkopen gaan doen en dat je steeds vaker terug ben komen.

8. **for account, group in sales14.groupby('accountName')["priceInEUR"]:**
   if group.shape[0] >= 2:
   trans_growth[account] = group.head(2).sum() < group.tail(2).sum()
   customers["trans_growth"] = pd.Series(trans_growth)
   Hier doen we de eerste twee aankopen bij elkaar tellen en de laatste twee. Als de laatste twee aankopen bij elkaar groter is dan de eerste twee aankopen. Dan betekent het dat je over tijd steeds een groter bedrag uitgeeft.
   True en False waarde zetten we uiteindelijk om in 1 en 0.

Bij het voorspellen kan je altijd te maken krijgen met overfitting. Overfitting houdt in dat je analyses precies of nauw overeenkomt met bepaalde set gegevens. Het faalt dan als je gegevens toevoegt of probeert te voorspellen.

Men heeft verschillende opties om overfitting te voorkomen (2017).

- **Cross-validatie**
  Hier ga je van je test data, kleinere test data maken. Dan test je de kleinere test data 's met zijn buren en zo kan je beter voorspellen.
- **Meer data gegevens gebruiken**
  Hoe meer data je hebt , hoe beter de voorspelling
- **Verwijderen van onnodige features**
  Dat kan je doen door te kijken wat voor toegevoegde waarden bepaalde feature heeft.
- ***"Early stopping"***
  Op tijd stoppen bij het verbeteren van je model. Op een gegeven moment wordt een model juist trager als je meer data toevoegt.
- **Ensembling**
  Het is een methode waarbij voorspellingen van verschillende lossen model samengevoegd wordt.

Als laatste voegen we natuurlijk een class toe aan tabel die kijkt of klanten idd zijn blijven inkopen doen. Daarvoor wordt de volgende functie gemaakt:

loyal = {}
customers["class"] = True
for account, group in customers.groupby('accountName')["class"]:

```
    if (account in sales15["accountName"].values):
        loyal[account] = True
    else:
        loyal[account] = False
customers["class"] = pd.Series(loyal)
customers["class"] = customers["class"].astype(int)
```

Hier wordt gecheckt of accountnamen voor komen in andere dataset.

Nu gaan we kijken naar de invloed van onze features door middel van volgende:

1. **Correlation matrix**
   Hier wordt de correlatie van de features op elkaar gekeken.
2. **Chi2 selector**
   De score uit deze functie selecteert de features met de hoogste waarden voor de test chi-squared statistiek.
3. **Scilearn inbuilt feature importance classifier**
   Dit is een functie die wederom kijkt welke feature het beste is om mee te voorspellen, zodat je een beter model kan maken

Nu gaan we *machine learning* algoritmes toepassen. Wij hebben gekozen voor 3 algoritmes namelijk:

1. **Logistic regression**
   Bij logistic regression kijkt die of iets juist of onjuist is. Bijvoorbeeld door middel van lengte en gewicht kan je voorspellen of iemand wel aan overgewicht lijdt of niet. Bij lineaire regressie kan je bijvoorbeeld wel voorspellingen doen als iemand dit gewicht zou hebben, wat zou dan de lengte kunnen zijn. Je kan hier door kijken welke features je het beste kan gebruiken, want soms door het toevoegen van features die geen toegevoegde waarde hebben, je model juist tegen werken, waardoor je verkeerde voorspellingen kan krijgen.
2. **Decision tree**
   Bij Decision trees wordt de data gesplitst in subsets. Omdat men met een gegeven niet kan weten of een klant wel of niet terugkomt. Bijvoorbeeld als het regent speel je soms wel buiten en soms niet. Dat hangt dan ook af van hoe hardt het waait en ook nog is wat voor temperatuur het is. Als je alleen zou weten of het regent dan weet je niet of men wel of niet buiten speelt. Daarom ga je data splitsen zodat je pure data subset krijgt. Waarbij een gegeven het duidelijk maakt of men wel of niet buiten speelt.
3. **Support vector machines**
   SVM brengt een lineaire scheiding aan tussen features die de features zo goed mogelijk van elkaar scheidt

Resultaten

Nadat je alle gegevens in een tabel hebt gezet krijg jet het volgende tabel.

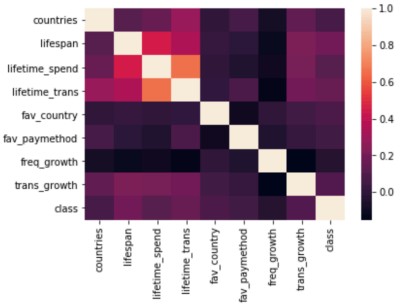
Dit zijn bijvoorbeeld de eerste 10 klanten:

accountName	countries	lifespan	lifetime_spend	lifetime_trans	fav_country	fav_paymethod	freq_growth	trans_growth	class
000113b526fbc8d19f03fccb9	1	5725342.0	14.000	3	1	1	0	0	0
0002c9dd969f5e5831d492524	1	7205755.0	40.000	5	1	0	0	0	0
000697979068d5bbc49dbbd64	1	0.0	10.000	1	0	0	1	0	0
00081bff5faac28727d229ab2	1	0.0	10.000	1	0	1	1	0	0
000b49aef4fdca6601be00775	1	3879836.0	30.000	3	1	1	0	0	1
000ba55fec599714628aa4e3b	1	0.0	20.000	1	1	1	1	0	1
000e9e31fa555ca178e7ab025	2	4342513.0	26.900	15	1	1	0	0	0
000f831d32866742c0c93e1e7	1	0.0	10.000	1	0	1	0	0	0
001879062f1b4dea3719de128	1	0.0	25.000	1	1	0	1	0	0
0018ef6545d446d90df80a0d0	1	2518250.0	7.000	2	0	1	0	0	0

De **correlatie matrix** geeft volgende tabel:

We merken natuurlijk dat elke feature perfect gecorreleert is op zich zelf. Er is geen duidelijke feature die veel zegt over class. We kunnen wel zien dat hoe meer je uitgeeft ook sterk gecorreleerd is met aantal totale tranacties die je hebt gedaan.

	countries	lifespan	lifetime_spend	lifetime_trans	fav_country	fav_paymethod	freq_growth	trans_growth	class
countries	1.000000	0.120854	0.161716	0.294982	0.001643	0.070054	-0.077231	0.143533	0.074937
lifespan	0.120854	1.000000	0.452776	0.344214	0.018446	-0.010099	-0.106917	0.215608	0.195295
lifetime_spend	0.161716	0.452776	1.000000	0.654994	0.012161	-0.038620	-0.104264	0.209986	0.123900
lifetime_trans	0.294982	0.344214	0.654994	1.000000	0.015811	0.087585	-0.136387	0.196263	0.157330
fav_country	0.001643	0.018446	0.012161	0.015811	1.000000	-0.102098	0.002590	0.059717	0.084557
fav_paymethod	0.070054	-0.010099	-0.038620	0.087585	-0.102098	1.000000	-0.040901	0.018092	0.052796
freq_growth	-0.077231	-0.106917	-0.104264	-0.136387	0.002590	-0.040901	1.000000	-0.149283	-0.032929
trans_growth	0.143533	0.215608	0.209986	0.196263	0.059717	0.018092	-0.149283	1.000000	0.105978
class	0.074937	0.195295	0.123900	0.157330	0.084557	0.052796	-0.032929	0.105978	1.000000



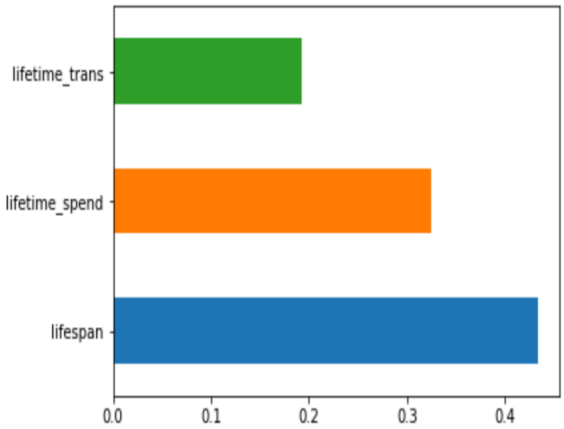
	Specs	Score
1	lifespan	4.658707e+10
2	lifetime_spend	1.954931e+05
3	lifetime_trans	4.831898e+04

Chi2 selector

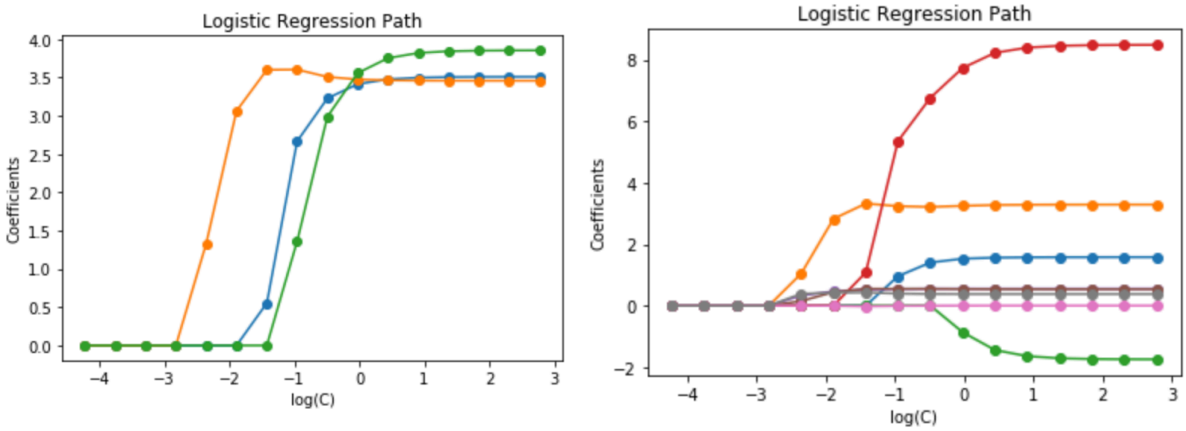
Hier zien we dat lifespan, lifetimes_spend en lifetime_trans als beste 3 features worden gekozen op basis van chi2 selector.

Scilearn inbuilt feature importance classifier

Uit deze grafiek zien we weer dat lifetime-trans, lifetime_spend en lifespan als beste drie features naar voren komen.

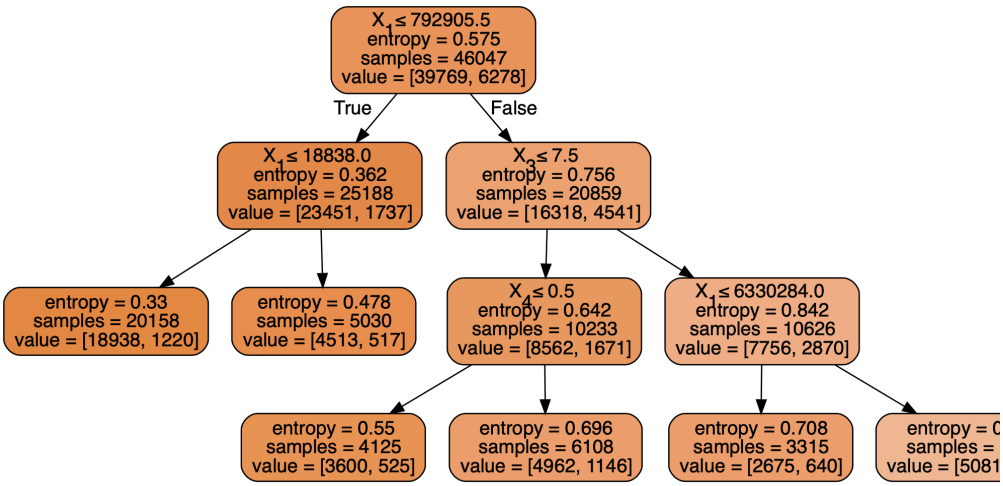


Logistic Regression Path

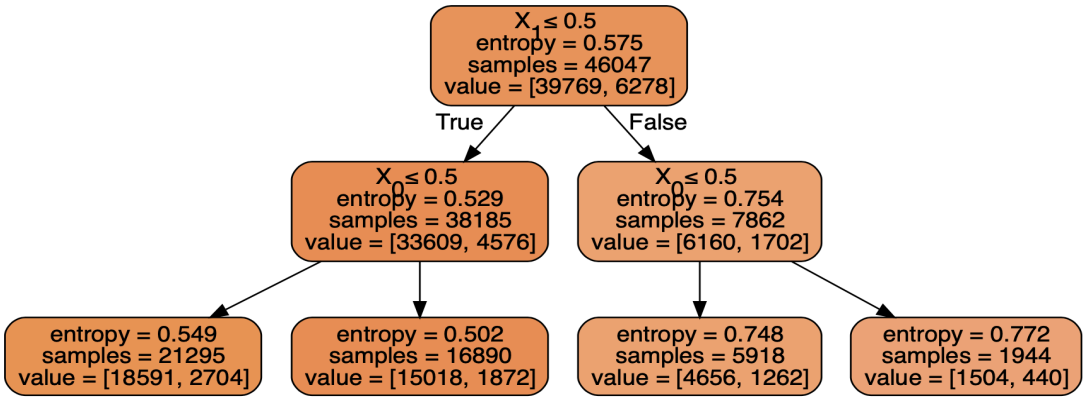


Hier boven je zie je 2 logistic regression path. Bij de linker zie je dat er 3 features zijn toegevoegd en bij rechter zie je dat alle 8 features zijn toegevoegd. Zo zie je dat voor het voorspellen of een klant terugkomt je niet alle features moet gebruiken.

Decision tree



Hierboven zie je bijvoorbeeld dat je met de features die wij hebben gekozen niet snel kan weten of een klant wel of niet blijft komen. Want je kan een max kiezen met het aantal keer dat er gesplitst kan worden en dan wordt het heel lange decision tree.



Hier zie je dan bijvoorbeeld wel als je een paar features neemt dat je sneller tot een beslissing kan komen. Daarom is het heel belangrijk om te kijken welke features je kiest voor je model. Veel features toevoegen aan je model kan je model juist slechter maken.

Support vector machines

```
In [24]: clf.predict(customers.iloc[1:100,0:4])

Out[24]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

Hier wordt er voorspelt voor eerste honderd klanten of ze terugkomen op basis van de eerste vier features. Zo kan je de features aanpassen en weer voorspellen en kan je dan kijken of het werkelijk zo is. Zo kan je bepalen hoe vaak het model juist voorspelt.

Conclusie

Op basis van onze onderzoeken kunnen we onze hypothese bevestigen:

Gedrag van klanten kan geanalyseerd worden door te gebruik maken van datatechnieken om beslissing te nemen om omzet te kunnen vergroten.

Hoewel we op basis van onze features moeilijk kunnen voorspellen of een klant wel of niet blijft komen. We kunnen wel zeggen dat we met deze features niet kunnen voorspellen en we betere features moeten kiezen. Daarnaast is het ook moeilijk om met een feature tot een besluit te komen, maar we weten nu ook dat we met meerdere features ook een slechter model krijgen. Hiervoor is het belangrijk dat we de juiste features kiezen en features die niet nodig zijn uit onze model verwijderen. Daarbij is het voor dat je een algoritme kiest al een juiste indeling maken van je data. En op basis van je data eerst een simpel model maken, zodat je een idee hebt waar je ongeveer naar toe wilt werken. Het is dus wel zo dat je klanten gedrag kan analyseren door gebruik te maken van data en daarop je beslissingen te nemen. Het is daar wel belangrijk dat je juiste proces volgt om een betrouwbaar model te krijgen.

Discussie

Wij hebben natuurlijk laten zien dat je met deze features geen beslissing kan nemen. Men wilt juist beter voorspellen en daarvoor moeten we nieuwe features kiezen en kijken of we op een andere manier wel een model kunnen maken die beter voorspellingen kan doen.

Daarnaast omdat sommige features van ons moeilijk te berekenen was, hebben we voor een simpele manier gekozen. Hierdoor komt de data ook niet overheen met de verkoopdata waardoor je ook moeilijke voorspellingen kan doen.

Ook aantal transacties is sterk gecorreleerd met totale bestedingen, daardoor als aantal transacties een effect heeft op de uitkomst dan heeft totale beslissing ook een effect.

Bibliografie

(2017, 09 07). Opgehaald van https://elitedatascience.com/overfitting-in-machine-learning

Takes, D. F. (2019). Opgehaald van http://liacs.leidenuniv.nl/~takesfw/DSPM/assignment2.html

Huang, B. Q., Kechadi, M. T., & Buckley, B. (2009, August). Customer churn prediction for broadband internet services. In International Conference on Data Warehousing and Knowledge Discovery (pp. 229-243). Springer, Berlin, Heidelberg.

Onditi, A. A. (2013). Relationship between customer personality, service features and customer loyalty in the banking sector: A survey of banks in Homabay County, Kenya.

Verhoef, P. C., & Franses, P. H. (2003). Combining revealed and stated preferences to forecast customer behaviour: three case studies. International Journal of Market Research, 45(4), 1-8.