In dit rapport gaan we proberen het gedrag van de klanten te voorspellen. We hebben tot onze beschikking de data van verkopen. Voor dat we voorspellingen kunnen doen om klant gedrag te kunnen analyseren, maken we een nieuwe tabel waar we de verkoopdata omzetten naar klanten data. Dat doen we door 8 *features* af te leiden van de verkoopdata die ons kunnen helpen om klant gedrag te kunnen meten.

De *feature* gaan ons helpen om te kijken of de klant terug blijft komen. Acht *feature* 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.

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?

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 inkoop

7. Transacties. Blijven mensen die steeds meer aankopen doen ook terugkomen.

8. Besteding. Blijven mensen die een groter bedrag uitbesteden doen ook terugkomen.

Nu gaan we uitleggen hoe we de functies hebben uitgewerkt.

1. **alcustomers['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.

1. **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 weegegeven gezien er soms maar seconden verschil zit.*

1. **customers['lifetime\_spend'] = sales14.groupby("accountName")["priceInEUR"].sum()**

*Hier pakken we het totale bedrag per klant.*

1. **customers['lifetime\_trans'] = sales14.groupby("accountName")["saleId"].nunique()**

*Hier pakken we het aantal transacties per klant.*

1. **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)*

1. **tempy = sales14.groupby("accountName")["methodId"].value\_counts()**

**temp6 = tempy.groupby("accountName").idxmax()**

*Dit pakken we op hetzelfde manier aan als bij 5.*

1. **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.*

1. trans\_growth = {}

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)

customers["trans\_growth"] = customers["trans\_growth"].fillna(0)

customers["trans\_growth"] = customers["trans\_growth"].astype(int)