PCA APPLICATION CREATING RECOMMENDER ENGINE

HIP SHOES

$95.00 USD

QUANTITY

− 1 + ADD TO CART

Not content with two rounds of New Balance 577 colabs, Erik and Peter at Sneakersnstuff have belted out Round 3 with another two contrasting colourways. Peter keeps to a classic black/white/3M design juxtaposing Erik's explosion of colour. The standout piece on both shoes is that for the first time, New Balance have allowed a designer to customize the ENCAP logo on the midsole to highlight the 'E' on Eric's shoe and the 'P' on Peter's. (sorry this is not shown on our pics, must be a sampling update)

Customers Who Bought This Item Also Bought

Cool Kicks

Saucony draws influences from the Caribbean Sea for their latest colourful Grid 9000 nicknamed 'Spr...
$90.95 USD

Gnarly Shoes

Nike have London on their minds with the 2012 Games mere months away. Following up from the gold me...
$100.00 USD

Gnarly Shoes

New silhouettes abound in 2016, the newest of which is this '90s sampling Zoom Toranada. With a foo...
$150.00 USD

Good Ol' Shoes

After some tantalising teaser pics, Michigan skate store Premier is set to drop their SB colab in t...
$115.00 USD

Insane Shoes

Rick Owens and adida you covered for your experimental high fasl sneaker needs all sum
$99.00 USD

INTRO

Data used - AdventureWorks DW2012 from

https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks

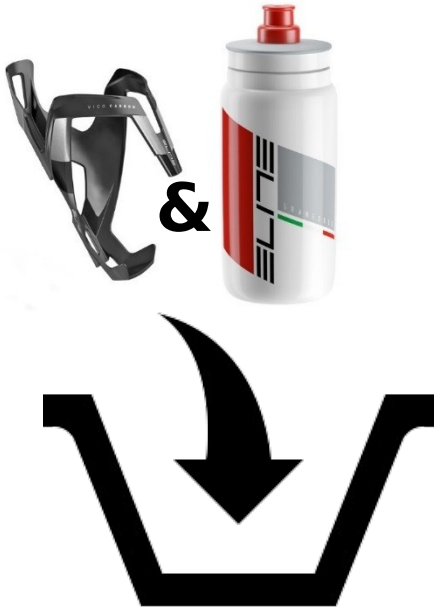
- AdventureWorks DataWarehouse is a Microsoft product sample. Data is from a fictitious, multinational manufacturing company called Adventure Works Cycles.
- Data warehouse backup to SQL Server
- Connect jupyter notebook to SQL Server
- Join desired tables

Principal Component Analysis

http://setosa.io/ev/

ITEM SIMILARITY

- Goal to derive correlations between items which are bought together - item similarity based on orders;
- If all orders are used to derive correlation between each item using for example Pearson correlation – this would take a substantial amount of computing power and would not perform well on large datasets;
- Thus, orders can be grouped together to represent latent/ generalised orders;
- Then a relationship between items can be derived using these grouped orders.

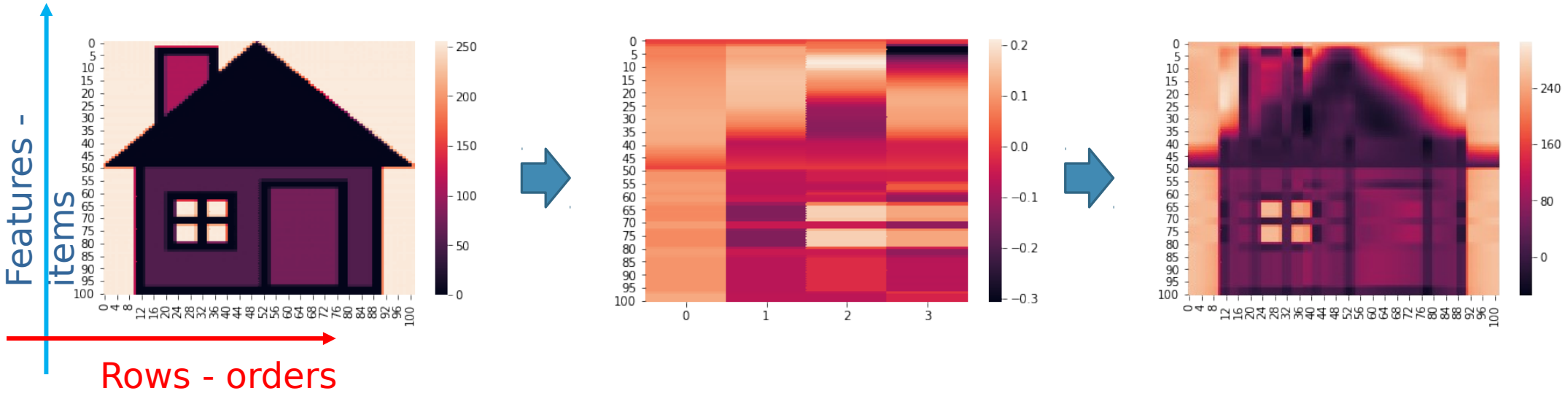


DATA PREPARATION FOR RECOMMENDER ENGINE

- 27 659 orders are contained within the current database;
- Orders made before 2007-07-01will be excluded from the model as the shopping behaviour and sold items have changed significantly since that date. This means that 20% of all orders will be removed;
- Colour and size removed from item title to improve model generalization (reduced from 130 to 40 products);
- Utility matrix comprised of orders (rows) and bought items (columns) was created. The intersection of particular order and item contains number of that particular item bought.

Product	AWC Logo Cap	All-Purpose Bike Stand	Bike Wash - Dissolver	Classic Vest	Fender Set - Mountain
SalesOrderNumber					
SO75119	0	0	0	0	0
SO75120	1	0	0	0	1
SO75121	0	0	0	0	0

PCA – TO GROUP ORDERS



- Let’s group rows by features using Principal Component Analysis (PCA);

- All the rows have been grouped together into 4 latent (hidden) groups;
- These 4 latent groups are preserving 90% of variation in data;

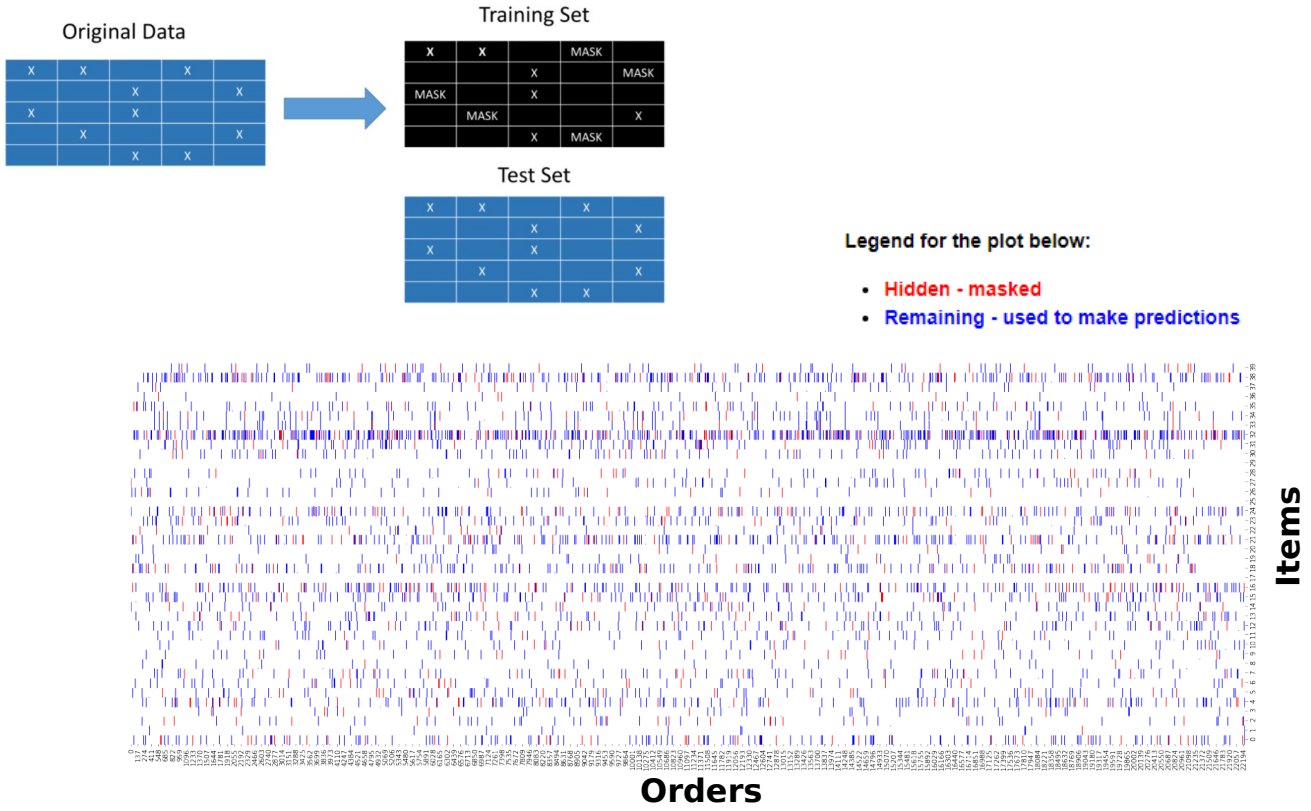
- Reconstructed features vs. rows matrix;
- Loss in resolution but basic pattern has been preserved;

PCA – TO GROUP ORDERS

```
01. import pandas as pd
02. import matplotlib.pyplot as plt
03. import seaborn as sns
04. import cv2
05. from sklearn.decomposition import PCA
06. 
07. img = cv2.imread('house.jpg',0)
08. df = pd.DataFrame(img)
09. 
10. sns.heatmap(img)
11. 
12. pca = PCA(n_components=0.9)
13. X_grouped = pca.fit_transform(df.T)
14. X_predicted = pca.inverse_transform(X_grouped).T
15. print(len(pca.components_))
16. 
17. sns.heatmap(pca.components_.T)
```

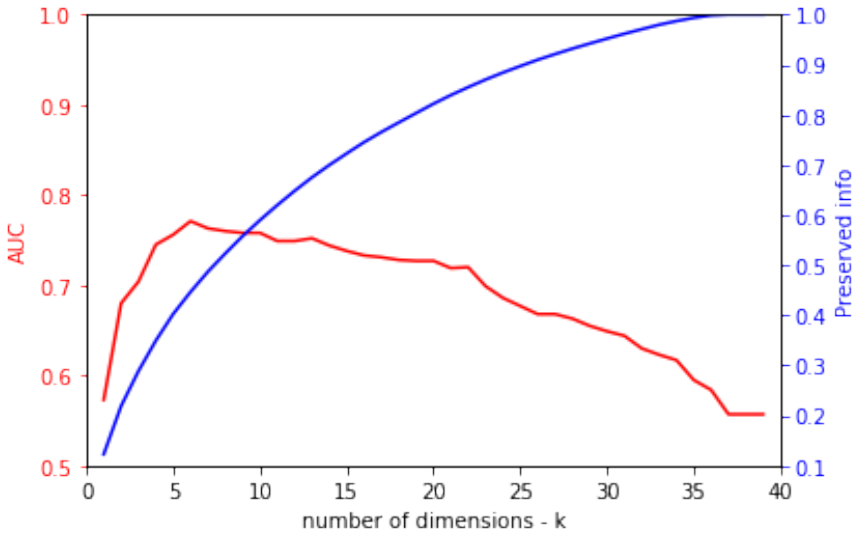
SPLITTING DATA INTO TRAINING AND TESTING

- Test data set – the same as original;
- Training data will be the same as original dataset but some item purchases will be deleted – masked;
- 20% of all items have been masked – hidden in the training set;



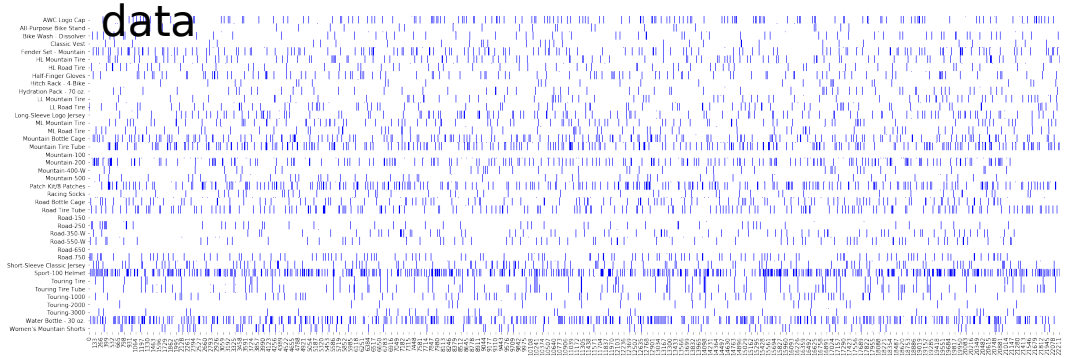
ORDER SIMILARITY

- How many latent/ underlying order groups should be chosen to depict underlying order pattern ?
- Receiver Operating Characteristic (ROC) curve was used to evaluate the performance of recommender engine with different number of latent order groups. This curve can was summarised in a single value for easier comparison of recommender engines performance by calculating the area under it **(AUC)**. AUC varies between 0.5 (random guesses) and 1.0 (everything guessed correctly).
- 6 groups are giving the best results (AUC=0.78). If more latent order groups are chosen, the model starts to overfit and not generalize well;
- AUC can be increased to 0.82 by “normalizing” data set – adding the number of products bought in total divided by the number of all orders for each product.

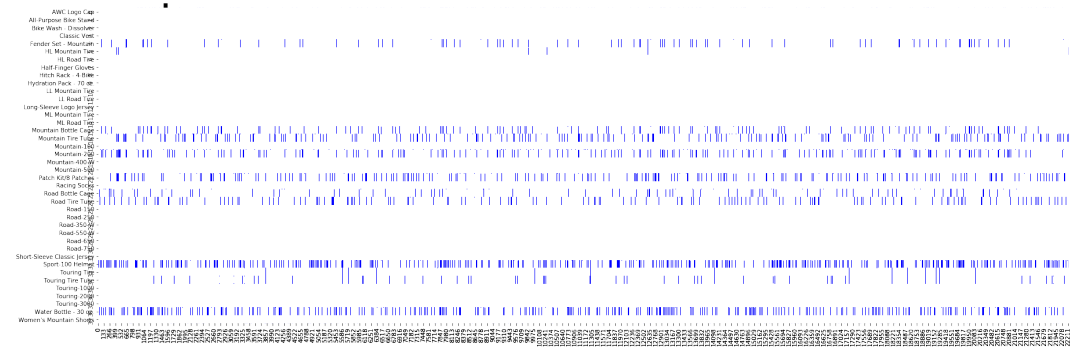


EXPLORATION OF 6 LATENT ORDER GROUPS

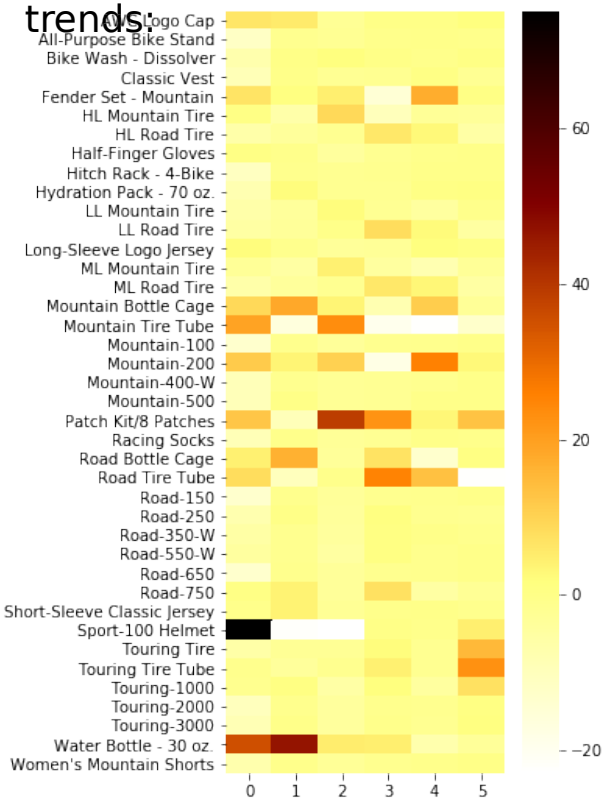
All orders – original data



All orders – reconstructed data using grouped



Latent order groups – shopping trends:



PEARSON’S CORRELATION

```
corr = np.corrcoef(X_grouped)

fig = plt.figure(figsize=(12,12))
ax1 = plt.subplot2grid((10, 10), (1, 0), colspan=9,rowspan=9)
ax2 = plt.subplot2grid((10, 10), (1, 9),rowspan=8)

sns.heatmap(corr, xticklabels=df.columns, yticklabels=df.columns,cmap="RdBu_r",square=True,ax=ax1,cbar_ax=ax2);
ax1.set_title("Pearson's Corelation Coefficient r",fontdict={'size':12, 'weight': 'bold'})
```

PEARSON’S CORRELATION

- Relationships between items will be defined using Person’s Correlation based on order groups/ trends derived using PCA;
- Pearson’s Correlation coefficient r (range -1 to +1):
  - -1.0 indicates strong negative linear relationship
  - +1.0 indicates strong positive linear relationship
  - 0.0 indicates that there is no linear relationship