Digital Technologies and Value Creation

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

Overview – subject to change

Overarching theme	Week	
Introduction	1	Introduction to analytics applications and coding basics
Gathering data	2	Scraping web data
Gathering data / descriptive analytics	3	Data pre-processing and descriptive analytics
Gathering data / descriptive analytics	4	Descriptives in marketing analytics, and using social media APIs
Descriptive analytics	5	Descriptives in people analytics
NO LECTURE	6	NO LECTURE
Predictive analytics	7	Retaining employees and customers with classification
Predictive analytics	8	Wrapping up classification and a deep-dive into dimensionality reduction
Predictive analytics	9	Segmenting customers and positioning products
Prescriptive analytics	10	Optimizing products and organizations
Prescriptive analytics	11	A/B-testing in practice

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Learning objectives of today

Goals:

- Understand the key differences between supervised and unsupervised learning (recap)
- Understand what dimensionality reduction is and how we measure how “useful” a feature is in a dataset
- Understand the ideas behind Principal Component Analysis and how to use it in practice

How will we do this?

- Activity: understanding the value of a feature
- Use of the McDonald’s menu to illustrate concepts (visually)

From supervised to unsupervised learning

Supervised learning is all about **prediction.**

Predicting a **value: Regression**	Predicting a **category: Classification**
Methods: Linear/polynomial regression, CART & related	**Methods**: Logistic regression, CART & related

Input to supervised learning:

$$(\boldsymbol{X}, \boldsymbol{y})$$

X: feature matrix (i.e., columns are features and rows are observations)

y: label (predicted value or category) for each observation

Unsupervised learning – no predictions!

Unsupervised learning **cannot do** prediction. Why?

Input to unsupervised learning:

$$\boldsymbol{X}$$

i.e., just the feature matrix – no labels, so no way of predicting.

What does unsupervised learning do?

Tries to **draw patterns or other information** out of X

What does unsupervised learning do?

This week: **Dimension reduction**

How can we construct, from our set of features, a smaller number of new features while keeping all the information?

Next week: **Clustering**

Can I group my data into similar groups?

Other possible topics: anomaly detection, association rules

“Issues” for unsupervised learning

- No simple goal for unsupervised learning: generally used in an **exploratory data analysis,** and as an **input to supervised learning**
- **No notion of “true answer”**: no immediate way of evaluating our work – often subjective
- As a consequence, no training/testing/validation/cross-validation possible

Dimensionality reduction with Principal Component Analysis

Example: McDonald's menu

The dataset we use is a list of elements on the McDonald's menu and their nutritional qualities given as percentage of daily recommended intake.

	Category	Item	Total Fat	Saturated Fat	Cholesterol	Sodium	Carbohydrates	Dietary Fiber	Vitamin A	Calcium	Iron	Vitamin C
0	Breakfast	Egg McMuffin	20	25	87	31	10	17	10	25	15	0
1	Breakfast	Egg White Delight	12	15	8	32	10	17	6	25	8	0
2	Breakfast	Sausage McMuffin	35	42	15	33	10	17	8	25	10	0
3	Breakfast	Sausage McMuffin with Egg	43	52	95	36	10	17	15	30	15	0
4	Breakfast	Sausage McMuffin with Egg Whites	35	42	16	37	10	17	6	25	10	0
...	...	...	...	...	...	...	...	...	...	...	...	...
254	Smoothies & Shakes	McFlurry with Oreo Cookies (Small)	26	44	14	12	27	4	15	40	8	0
255	Smoothies & Shakes	McFlurry with Oreo Cookies (Medium)	35	58	19	16	35	5	20	50	10	0
256	Smoothies & Shakes	McFlurry with Oreo Cookies (Snack)	17	29	9	8	18	2	10	25	6	0
257	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Medium)	50	76	20	17	38	9	20	60	6	0
258	Smoothies & Shakes	McFlurry with Reese's Peanut Butter Cups (Snack)	25	38	10	8	19	5	10	30	4	0

McDonald's dataset: 12 features, 10 of which are numerical.

What does PCA do?

- PCA only works on numerical datasets: 10 features here.
- It creates the **same number** of **new features by taking linear combinations** of the **old features centered**, e.g.,

$$z_1 = \boldsymbol{a_{01}} \cdot (\text{Total Fat} - \overline{\text{Total Fat}}) + \cdots + \mathbf{a_{91}} \cdot (\text{Vitamin C} - \overline{\text{Vitamin C}})$$
$$\vdots$$
$$z_{10} = \boldsymbol{a_{91}} \cdot (\text{Total Fat} - \overline{\text{Total Fat}}) + \cdots + \boldsymbol{a_{99}} \cdot (\text{Vitamin C} - \overline{\text{Vitamin C}})$$

scores **loadings**

How is this **dimension reduction** when we have the same number of features**?**

An activity to start with

Let’s understand how this does lead to dimension reduction through an activity [only on 3 features]

Complete Part 1 of the **Exercise notebook**

Old dataset

	Category	Item	Vitamin C	Total Fat	Cholesterol
0	Breakfast	Egg McMuffin	0	20	87
1	Breakfast	Egg White Delight	0	12	8
2	Breakfast	Sausage McMuffin	0	35	15
3	Breakfast	Sausage McMuffin with Egg	0	43	95
4	Breakfast	Sausage McMuffin with Egg Whites	0	35	16

New dataset

	Category	Item	z1	z2	z3
0	Breakfast	Egg McMuffin	56.768221	2.172171	-39.444570
1	Breakfast	Egg White Delight	-12.352350	-10.855976	-2.597695
2	Breakfast	Sausage McMuffin	5.910870	-7.809975	12.737682
3	Breakfast	Sausage McMuffin with Egg	75.851241	5.374492	-24.660107
4	Breakfast	Sausage McMuffin with Egg Whites	6.730670	-7.653655	12.186767

Reminder: covariance matrices

	Feature 1	Feature 2
Feature 1	Variance of Feature 1	Covariance of Features 1 & 2
Feature 2	Covariance of Features 1 & 2	Variance of Feature 2

```
menu.cov()
```

	Vitamin C	Total Fat	Cholesterol
Vitamin C	694.087600	-51.518681	-63.855331
Total Fat	-51.518681	478.961925	433.180814
Cholesterol	-63.855331	433.180814	846.324265

```
df.cov()
```

	z1	z2	z3
z1	1.147864e+03	7.440345e-07	-0.012614
z2	7.440345e-07	6.795013e+02	-0.001424
z3	-1.261391e-02	-1.424010e-03	192.011090

Total variance of **old** dataset:
$694.08 + 478.96 + 433.18 = \mathbf{2019.37}$

Total variance of **new** dataset:
$1147.86 + 679.50 + 192.01 = \mathbf{2019.37}$

The two datasets contain the **same amount of information**.

Activity recap

```
menu.cov()
```

	Vitamin C	Total Fat	Cholesterol
Vitamin C	694.087600	-51.518681	-63.855331
Total Fat	-51.518681	478.961925	433.180814
Cholesterol	-63.855331	433.180814	846.324265

```
df.cov()
```

	z1	z2	z3
z1	1.147864e+03	7.440345e-07	-0.012614
z2	7.440345e-07	6.795013e+02	-0.001424
z3	-1.261391e-02	-1.424010e-03	192.011090

Ratio Variance/Variance of total dataset:

Vitamin C: $\frac{694.08}{2019} \approx 34\%$

Total Fat: $\frac{478.96}{2019} \approx 24\%$

Cholesterol: $\frac{846.32}{2019} \approx 43\%$

Ratio Variance/Variance of total dataset:

z_1: $\frac{1147}{2019} \approx 57\%$

z_2: $\frac{478.96}{2019} \approx 34\%$

z_3: $\frac{846.32}{2019} \approx 9\%$

The two datasets contain the same amount of information but **spread out in a different** way among the features:

Old dataset: cannot drop features

New dataset: **drop** z_3 (9% of the information)

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Dimension reduction through PCA

PCA creates **new features $z_1, z_2, \ldots, z_n$** as linear combinations of old features such that:

- z_1 has the **highest variance possible**
- z_2 has the **highest variance possible** among all **variables with covariance = 0 with z_1**
- z_3 has the **highest variance possible** among all **variables with covariance = 0 with z_1, z_2** etc.

```
df.cov()
```

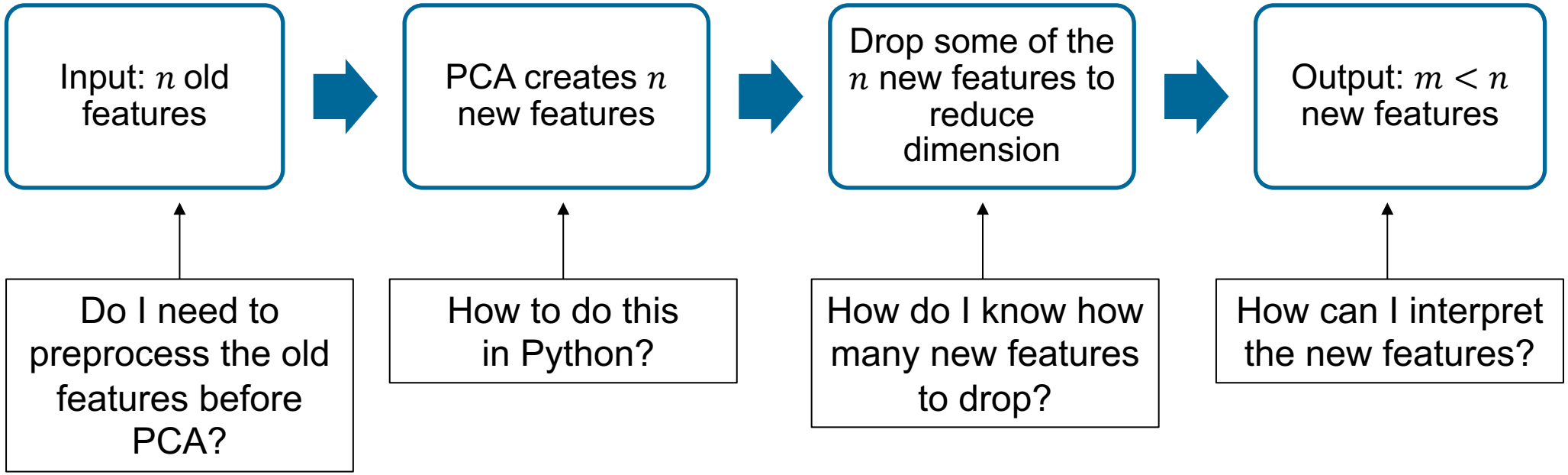
	z1	z2	z3
z1	1.147864e+03	7.440345e-07	-0.012614
z2	7.440345e-07	6.795013e+02	-0.001424
z3	-1.261391e-02	-1.424010e-03	192.011090

- The last new features have small variance
- Can be dropped without losing too much information

→ **Dimension Reduction**

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

PCA so far…



We answer these questions now.

Pre-processing and PCA in Python

Preprocessing the data

- PCA only works on **numerical datasets** with **no missing values**
  → one hot-encoding if needed, imputation

- What about **scaling (=centering +dividing by std dev)?**

Centering the old variables

Doesn't change anything whether it is done or not (so up to you).
Why?

Student	Weight (kgs) pre-MSc
1	65
2	90
3	82
4	71

Variance? 93.5

Student	Weight (kgs) post-MSc
1	63
2	88
3	80
4	69

Variance? 93.5

PCA based on variance & same variance

Dividing the old vars by std deviation

Necessary when the feature units are different/incomparable.

Student	Height (cm)
1	178
2	163
3	181
4	175

Student	Height (m)
1	1.78
2	1.63
3	1.81
4	1.75

Can lead to big differences in PCA results if not done

Variance? 46.6875

Variance? 0.0046

In McDonald's dataset: not needed as everything in %

Part 2 of the notebook

```
pca=dcp.PCA(n_components=3)
pca.fit(menu)
```

Running PCA (to get 3 new features)

```
pca.components_
```

```
array([[-0.17718634,  0.54454878,  0.81979975],
       [ 0.98405437,  0.08485918,  0.15631992],
       [ 0.01555629,  0.83442528, -0.55090149]])
```

Loadings

```
pca.explained_variance_
```

```
array([1147.86421744,  679.5013194 ,  192.00825288])
```

Variance of new feature

```
pca.explained_variance_ratio_
```

```
array([0.56842583, 0.33649111, 0.09508307])
```

Ratio variance of new feature/total variance

```
data_pca = pca.fit_transform(menu_num)
data_pca
```

Scores

Component choice and interpretation

Number of variables to include

We consider the full McDonald's dataset with all the features

	Total Fat	Saturated Fat	Cholesterol	Sodium	Carbohydrates	Dietary Fiber	Vitamin A	Calcium	Iron	Vitamin C
0	20	25	87	31	10	17	10	25	15	0
1	12	15	8	32	10	17	6	25	8	0
2	35	42	15	33	10	17	8	25	10	0
3	43	52	95	36	10	17	15	30	15	0
4	35	42	16	37	10	17	6	25	10	0
...	...	...	...	...	...	...	...	...	...	...
254	26	44	14	12	27	4	15	40	8	0
255	35	58	19	16	35	5	20	50	10	0
256	17	29	9	8	18	2	10	25	6	0
257	50	76	20	17	38	9	20	60	6	0
258	25	38	10	8	19	5	10	30	4	0

We can run PCA on this set of features.
How do we know **how many of them to keep out of the 10 we generate?**

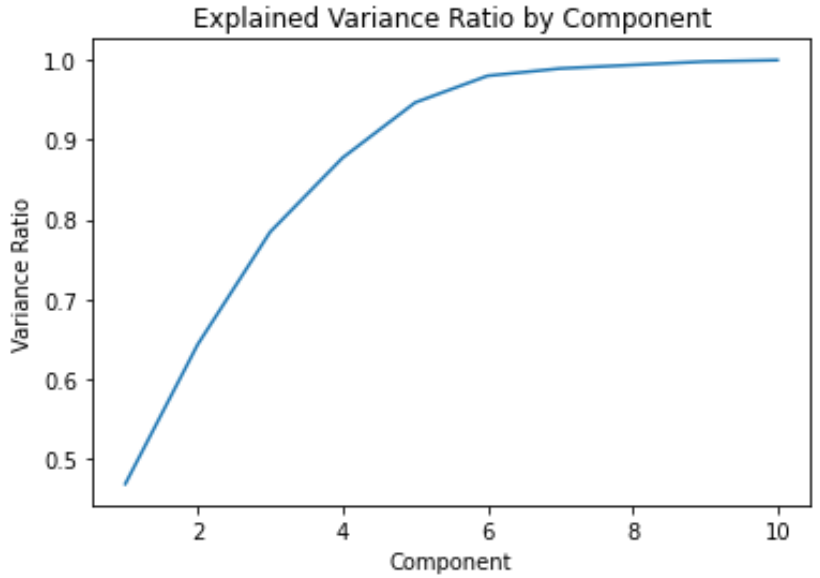
Number of variables to include

We run PCA on these features and get:

```
pca.explained_variance_ratio_
```

```
array([0.46846089, 0.17496823, 0.14112873, 0.09299599, 0.06924932,
       0.03351428, 0.00917541, 0.00442827, 0.00427006, 0.00180882])
```

We now plot the cumulative sum of the explained variance ratios:



Our goal is to meet a certain **threshold of explained variance:**

- More than 50%
- Even better: more than 80%

Here: 3-4 components would be good. Certainly not more than 5.

python™

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Interpreting the principal components

PCA can sometimes provide additional insights into the data.

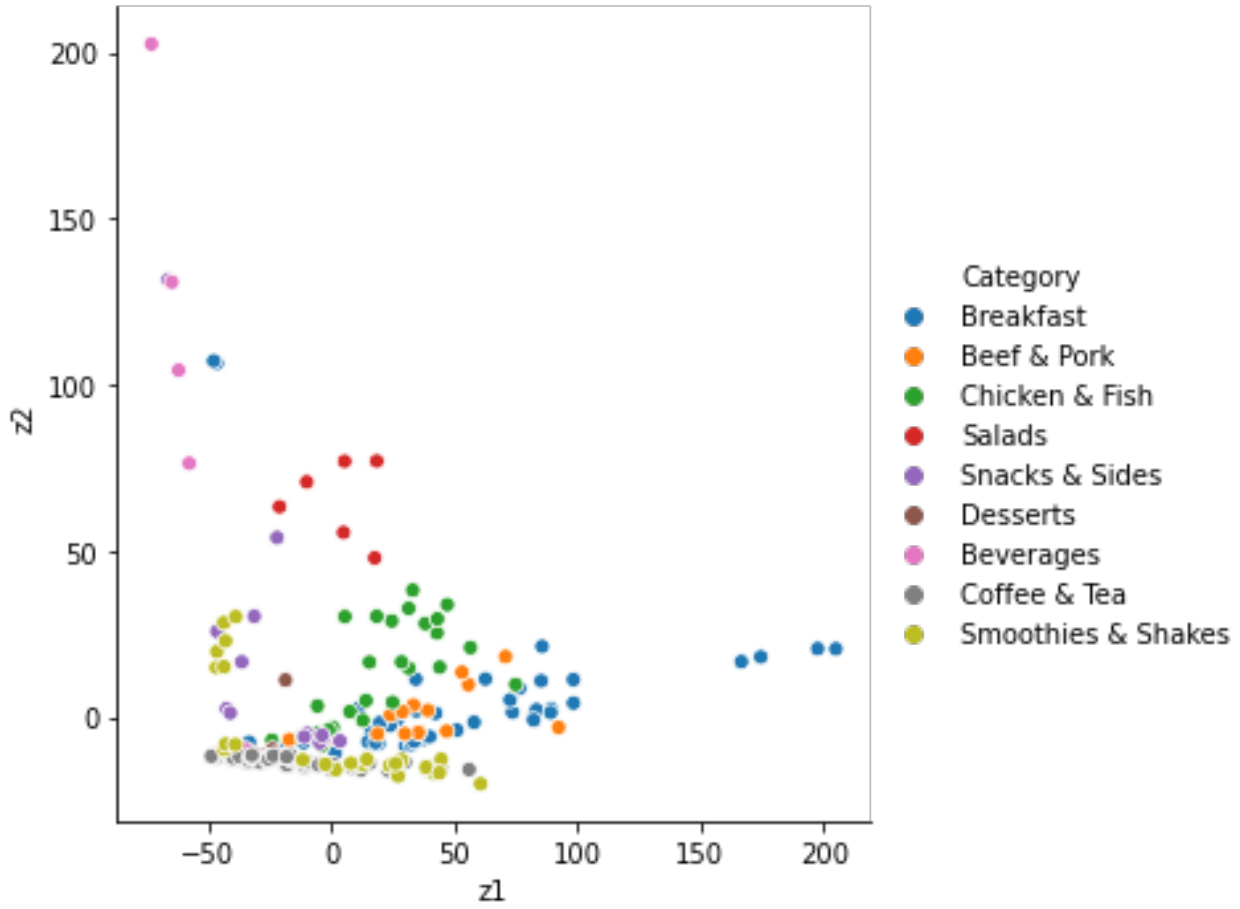
To find those, look at **loadings (sign and relative value):**

	Total fat	Saturated fat	Cholest.	Sodium	Carbs	Dietary Fiber	Vitamin A	Calcium	Iron	Vitamin C
z1	0.414	0.525	0.547	0.413	0.094	0.077	0.080	0.114	0.154	-0.120
z2	0.033	-0.105	0.100	0.155	-0.025	-0.087	0.325	-1.682	0.069	0.898

Here for example:

- the first component corresponds to high fat and high salt content
- the second component corresponds to high vitamin A and vitamin C

Interpreting the principal components



What do you feel is missing as a feature here?

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

PCA in the next group assignment

Boats – the case

- Mary is Senior Manager in the Customer Insights department of CreeqBoat.
- She has conducted market research around needs in the boating industry.

Overarching goal: using the study to

- **understand the market segments** that make up her potential client basis
- **understand the key purchase drivers** in each segment
- **develop a targeted strategy** based on the understanding of customer segments

Usual caveats:

- **Limited by time:** this generally would take weeks and require many different people. The assignment is consequently simplified and can be improved.
- **No one best answer:** many different answers informed by our backgrounds and levels of knowledge of the area. It is important to justify your reasoning!

Boats, Part 1: The key components that inform segmentation

Goal: Boil down the 29 (!!) questions about customer attitudes to a few key factors.

How to do this?

Use PCA!

- **How many components to keep?**
- **Association between original 29 variables and the components?**
- **Interpretation of components?**

Sparse PCA tries to compromise between two things:

- doing the “real” PCA
- pushing for a result where a lot of the loadings are zero

This is controlled by parameter alpha. If alpha=0, real PCA, if alpha large, a lot of zeros ⇒ take alpha=5.

Boats, Part 2: The actual customer segmentation

Goal: We now have the key factors that “make up” each client. We need to understand now if there are certain typical client profiles in terms of needs.

How to do this?

Use clustering (videos and next session)!

- **How many clusters to use?**
- **Check robustness of clusters**
- **Use Q2-Q16 to profile these clusters**
- **Develop a targeting strategy**

See you next week!