Introduction

DIMENSIONALITY REDUCTION IN PYTHON

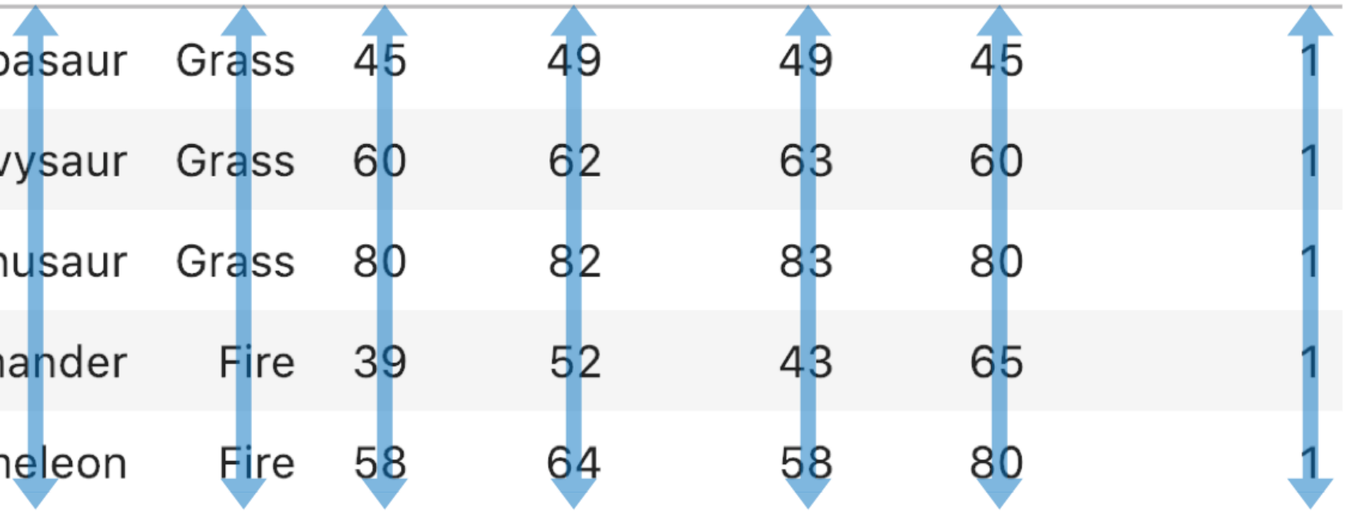
Jeroen Boeye
Machine Learning Engineer, Faktion

Tidy data

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

Tidy data

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1



Tidy data

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

Tidy data

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

The shape attribute

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

```
pokemon_df.shape
```

```
(5, 7)
```

When to use dimensionality reduction?

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

When to use dimensionality reduction?

Name	Type	HP	Attack	Defense	Speed	Generation
Bulbasaur	Grass	45	49	49	45	1
Ivysaur	Grass	60	62	63	60	1
Venusaur	Grass	80	82	83	80	1
Charmander	Fire	39	52	43	65	1
Charmeleon	Fire	58	64	58	80	1

The describe method

```
pokemon_df.describe()
```

	HP	Attack	Defense	Speed	Generation
count	5.0	5.0	5.0	5.0	5.0
mean	56.4	61.8	59.2	66.0	1.0
std	15.9	13.0	15.4	14.7	0.0
min	39.0	49.0	43.0	45.0	1.0
25%	45.0	52.0	49.0	60.0	1.0
50%	58.0	62.0	58.0	65.0	1.0
75%	60.0	64.0	63.0	80.0	1.0
max	80.0	82.0	83.0	80.0	1.0

The describe method

```
pokemon_df.describe()
```

	HP	Attack	Defense	Speed	Generation
count	5.0	5.0	5.0	5.0	5.0
mean	56.4	61.8	59.2	66.0	1.0
std	15.9	13.0	15.4	14.7	0.0
min	39.0	49.0	43.0	45.0	1.0
25%	45.0	52.0	49.0	60.0	1.0
50%	58.0	62.0	58.0	65.0	1.0
75%	60.0	64.0	63.0	80.0	1.0
max	80.0	82.0	83.0	80.0	1.0

The describe method

```
pokemon_df.describe(exclude='number')
```

	Name	Type
count	5	5
unique	5	2
top	Charmander	Grass
freq	1	3

The describe method

```
pokemon_df.describe(exclude='number')
```

	Name	Type
count	5	5
unique	5	2
top	Charmander	Grass
freq	1	3

Let's practice!

DIMENSIONALITY REDUCTION IN PYTHON

Feature selection vs feature extraction

DIMENSIONALITY REDUCTION IN PYTHON

Jeroen Boeye
Machine Learning Engineer, Faktion

Why reduce dimensionality?

Your dataset will:

- be less complex
- require less disk space
- require less computation time
- have lower chance of model overfitting

Feature selection

income	age	favorite color
10000	18	Black
50000	47	Blue
20000	40	Blue
30000	29	Green
20000	22	Purple

Feature selection

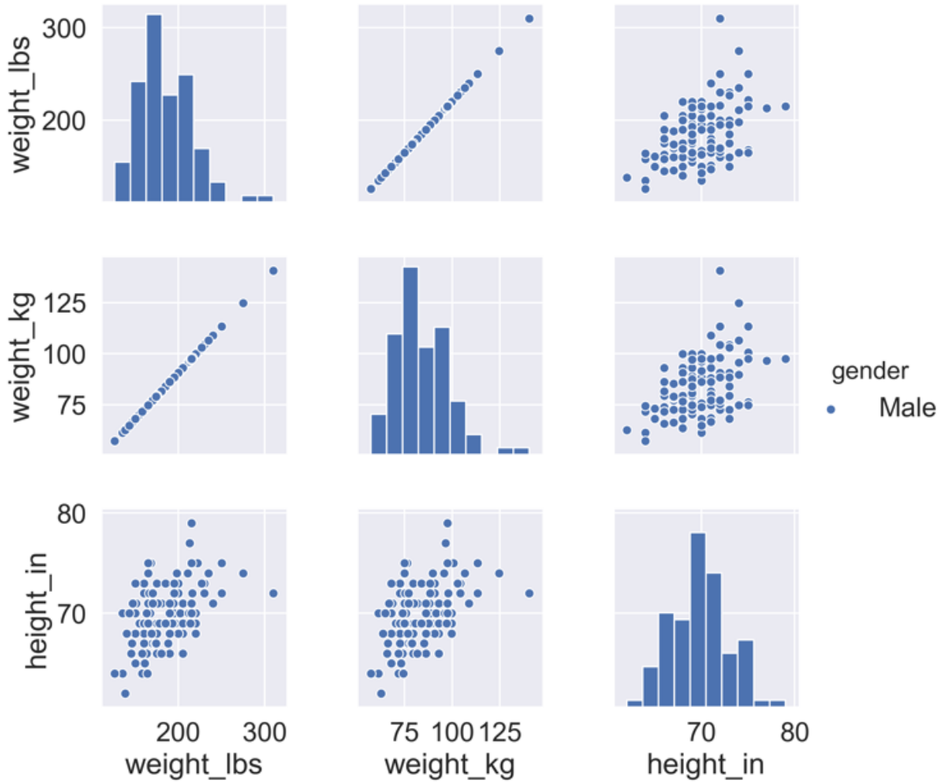
income	age	favorite color
10000	18	Black
50000	47	Blue
20000	40	Blue
30000	29	Green
20000	22	Purple

income	age
10000	18
50000	47
20000	40
30000	29
20000	22

```
insurance_df.drop('favorite color', axis=1)
```

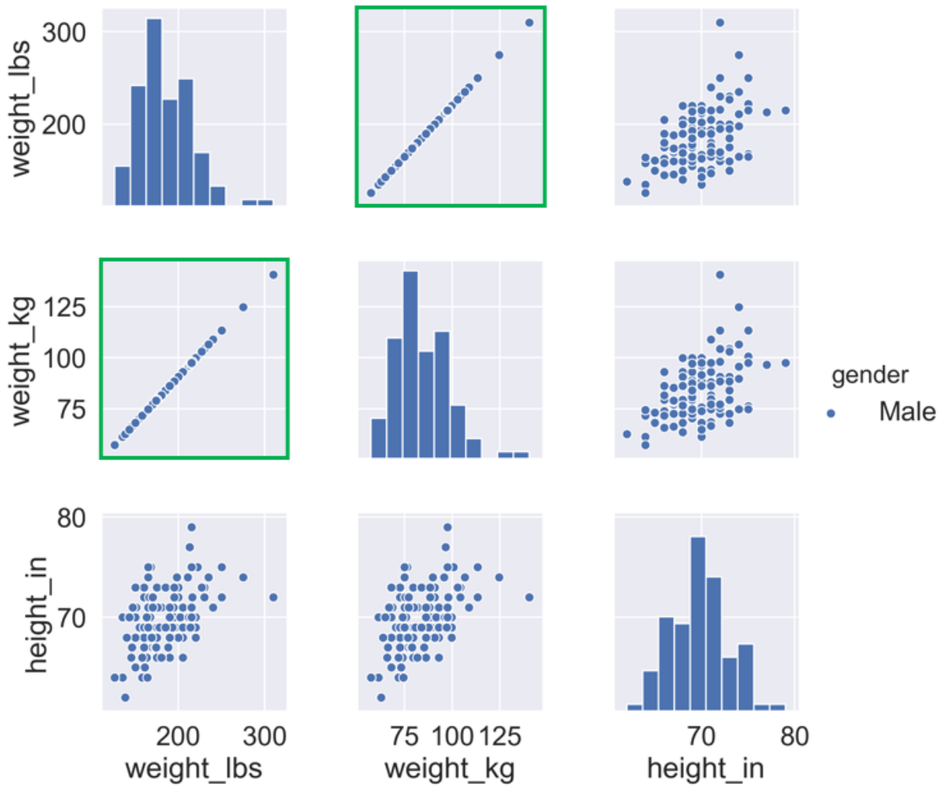
Building a pairplot on ANSUR data

```
sns.pairplot(ansur_df, hue="gender", diag_kind='hist')
```



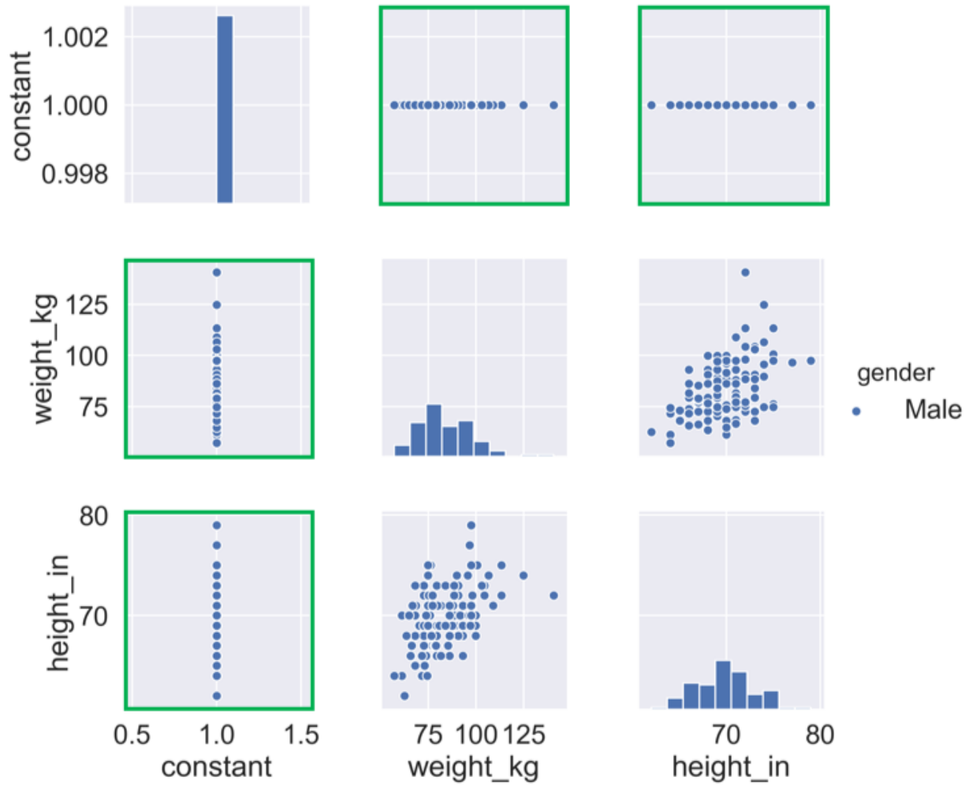
Building a pairplot on ANSUR data

```
sns.pairplot(ansur_df, hue="gender", diag_kind='hist')
```

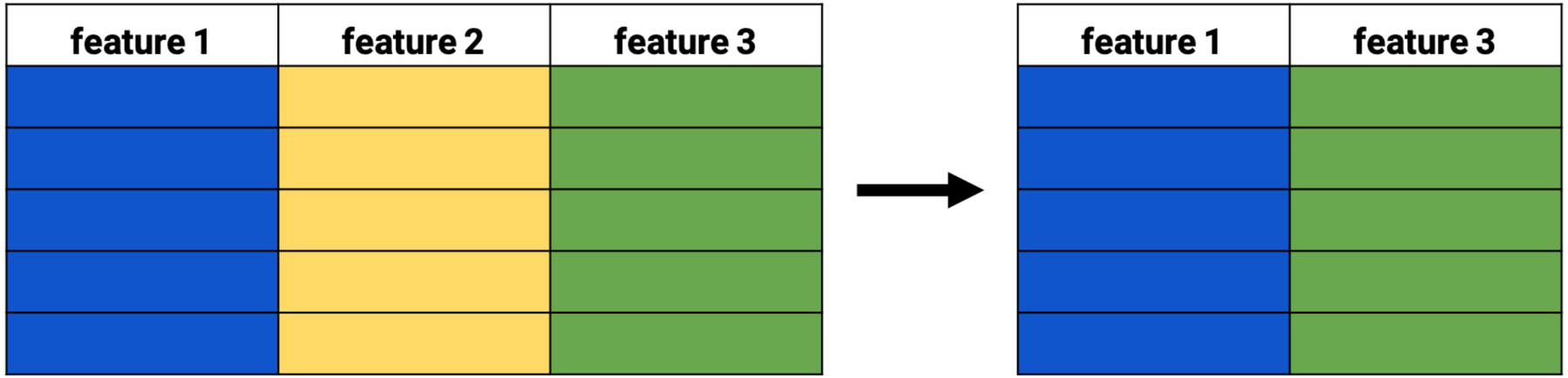


Building a pairplot on ANSUR data

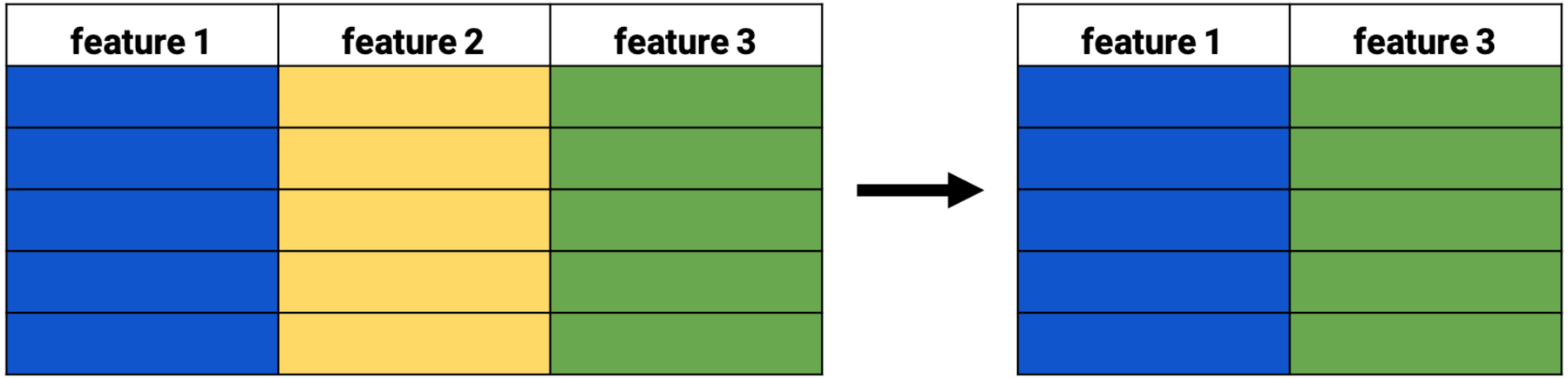
```
sns.pairplot(ansur_df, hue="gender", diag_kind='hist')
```



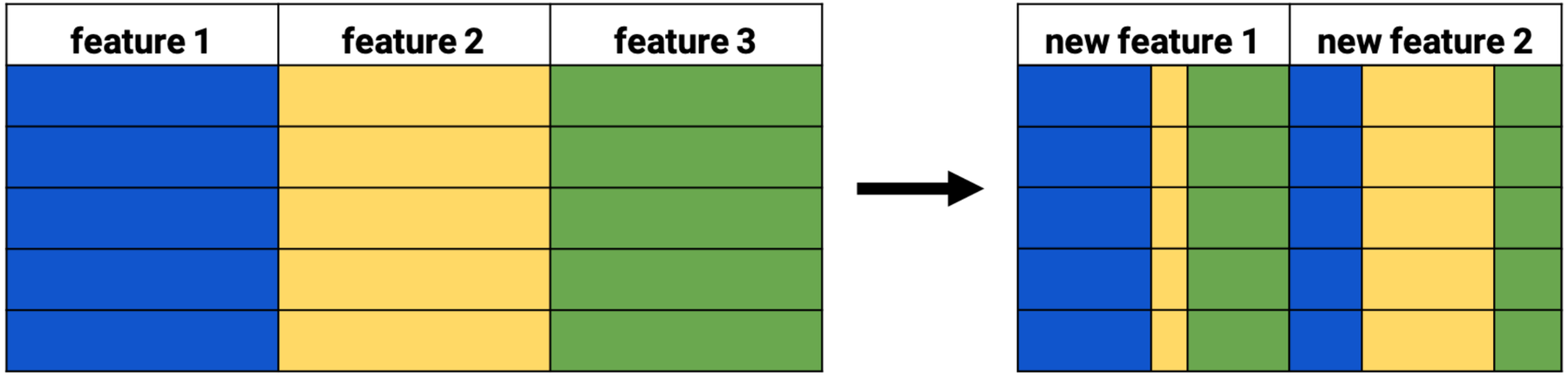
Feature selection



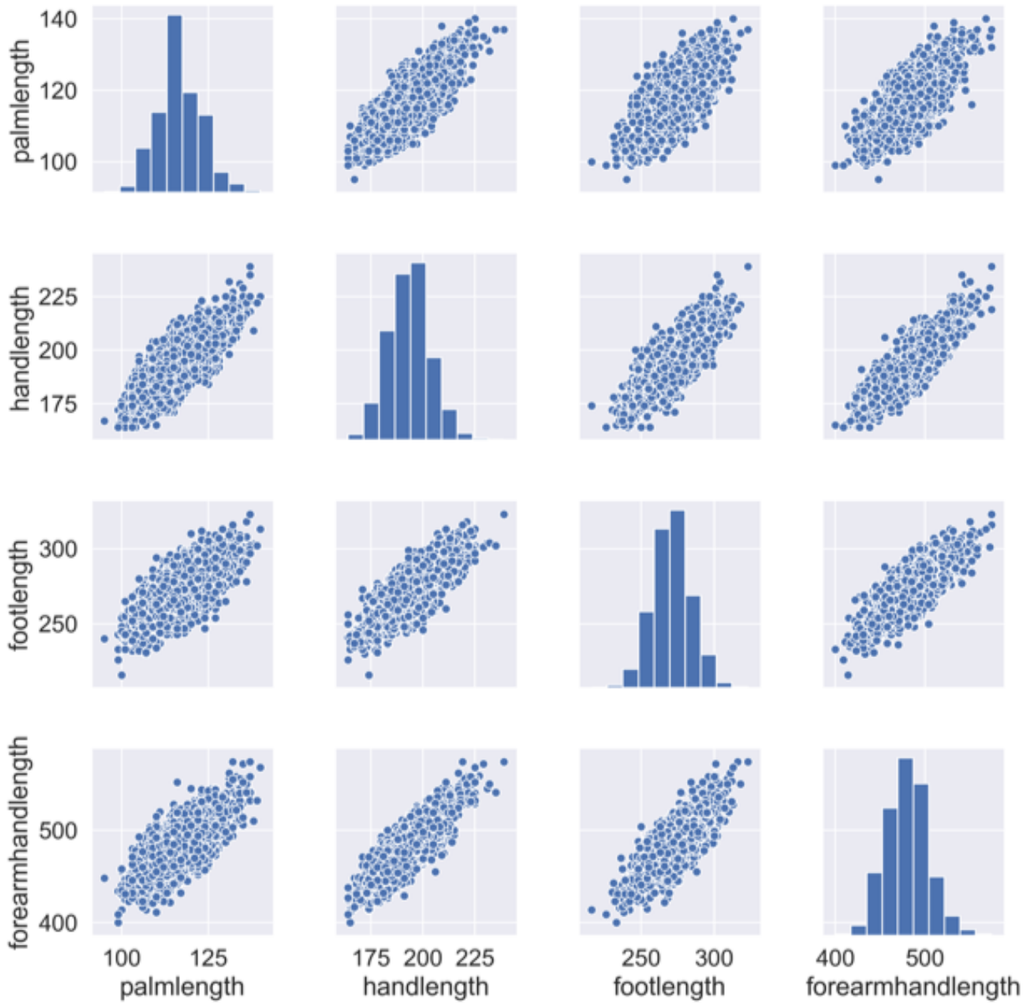
Feature selection



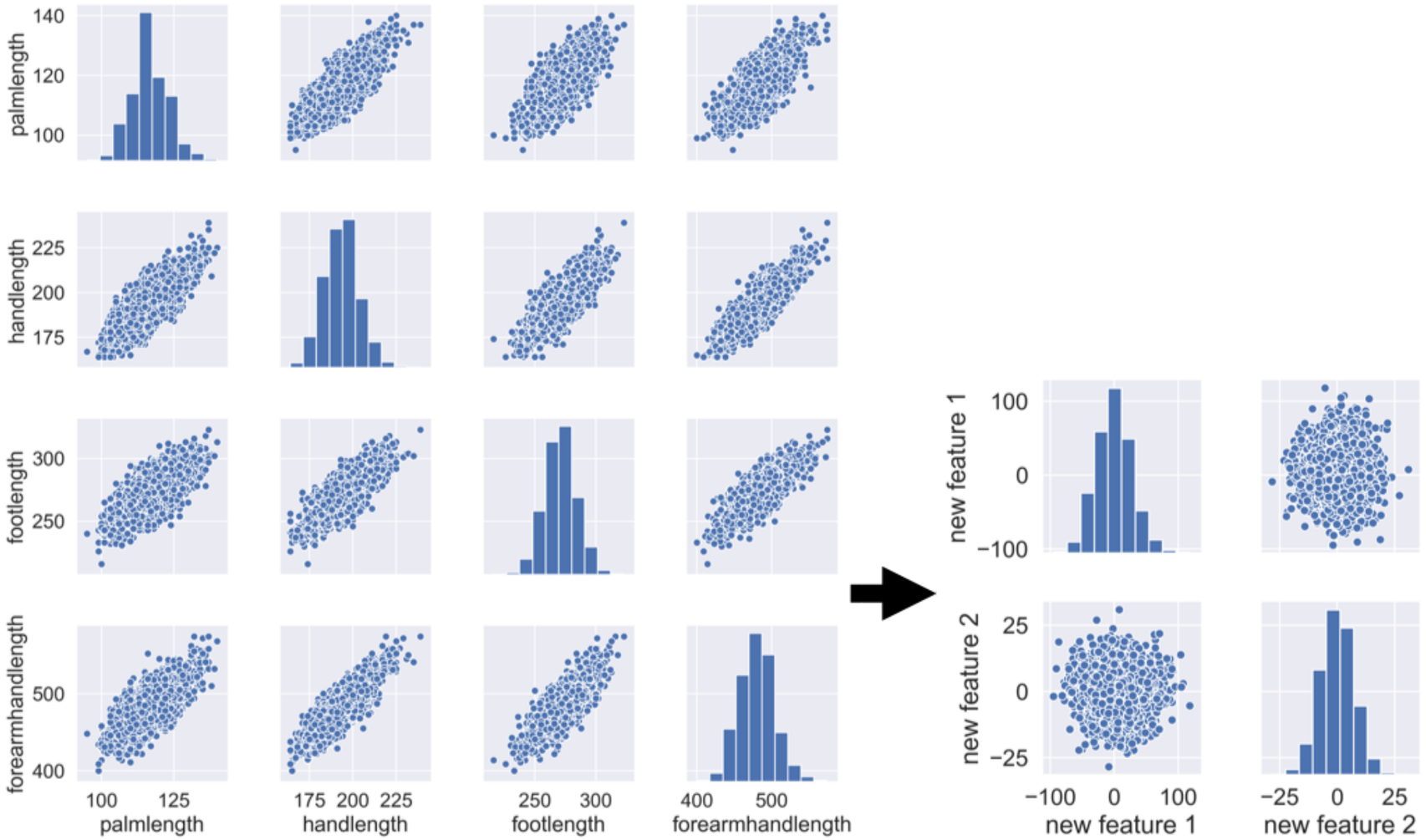
Feature extraction



Feature extraction - Example



Feature extraction - Example



Let's practice!

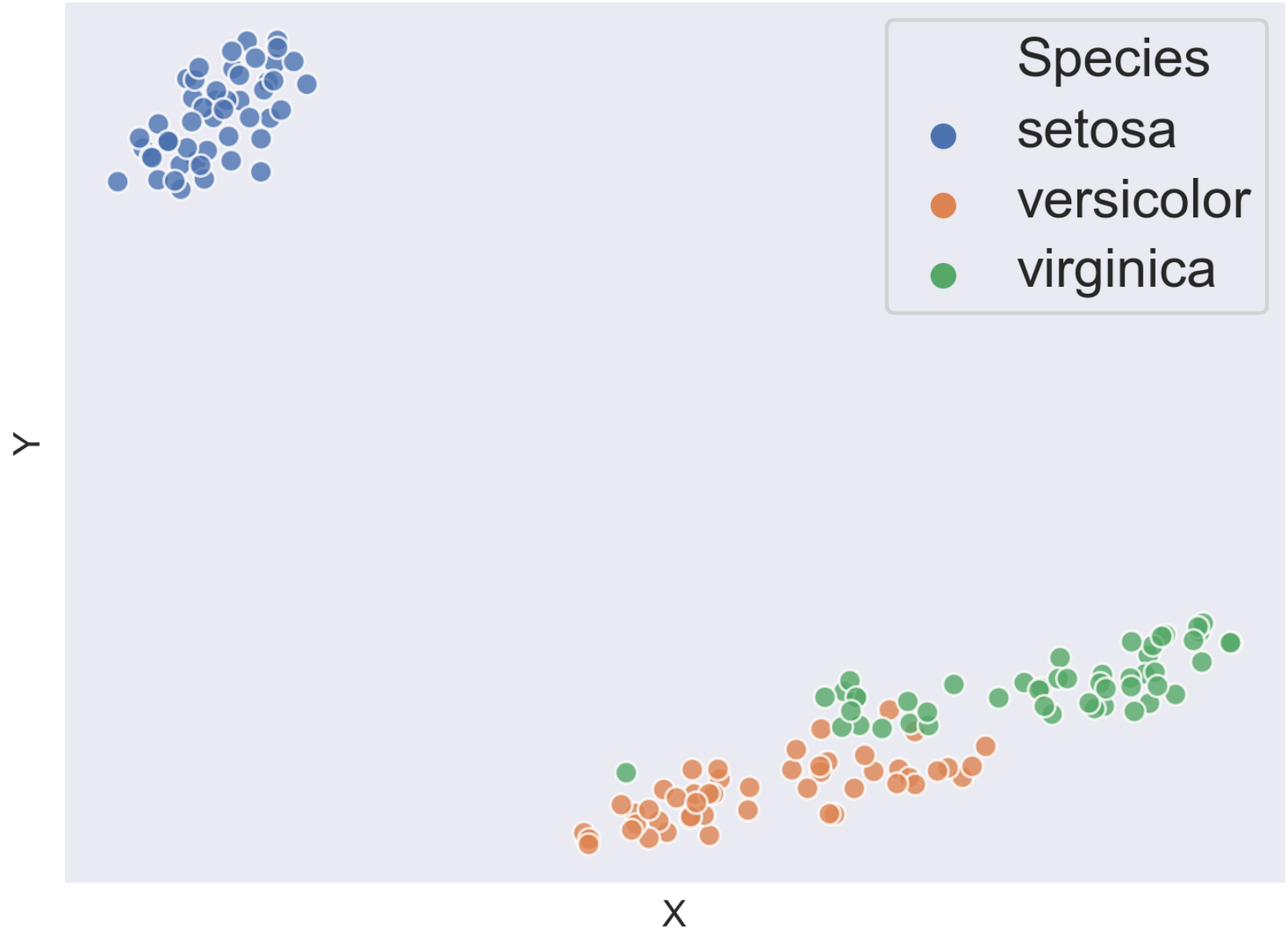
DIMENSIONALITY REDUCTION IN PYTHON

t-SNE visualization of high-dimensional data

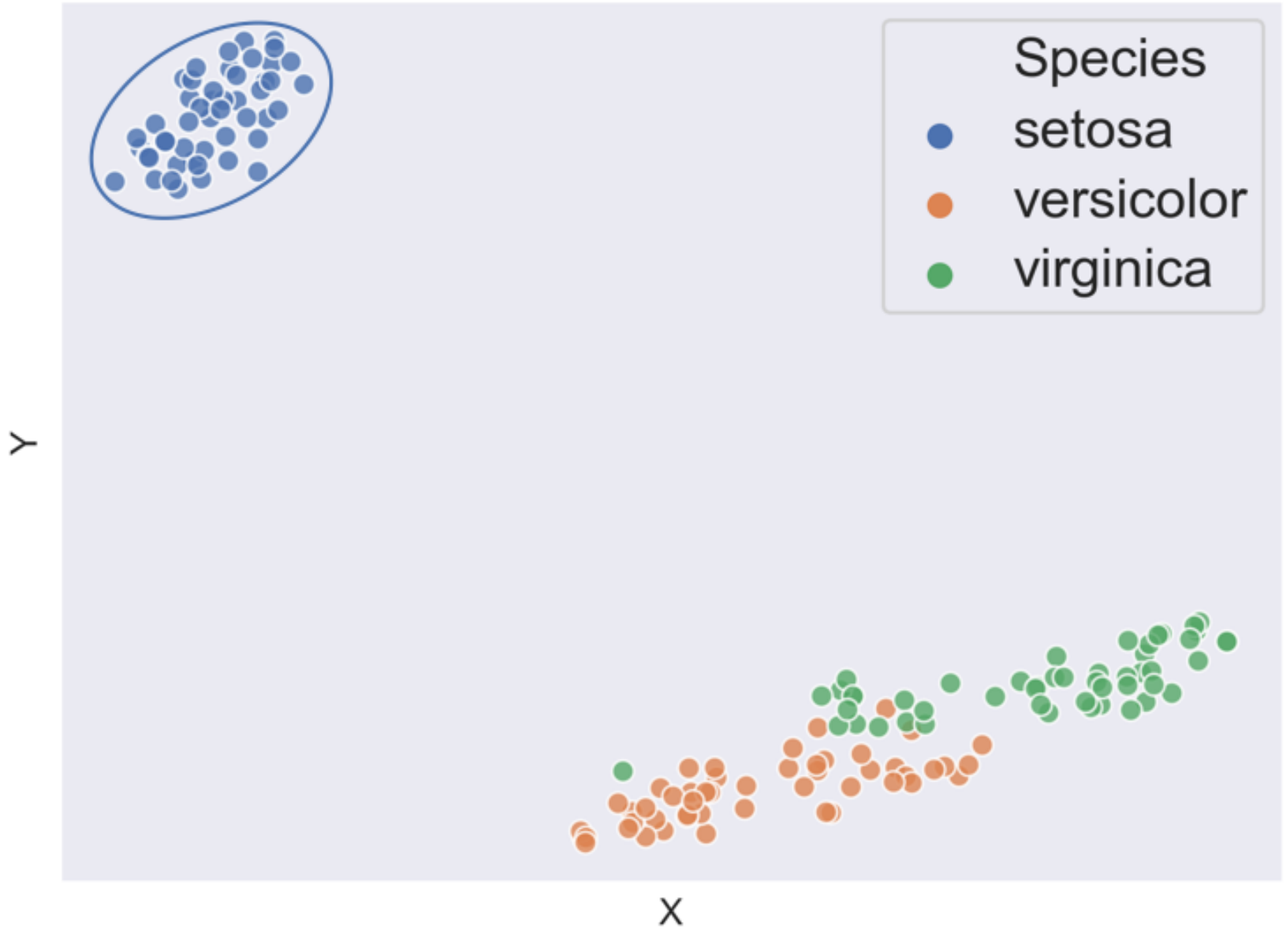
DIMENSIONALITY REDUCTION IN PYTHON

Jeroen Boeye
Machine Learning Engineer, Faktion

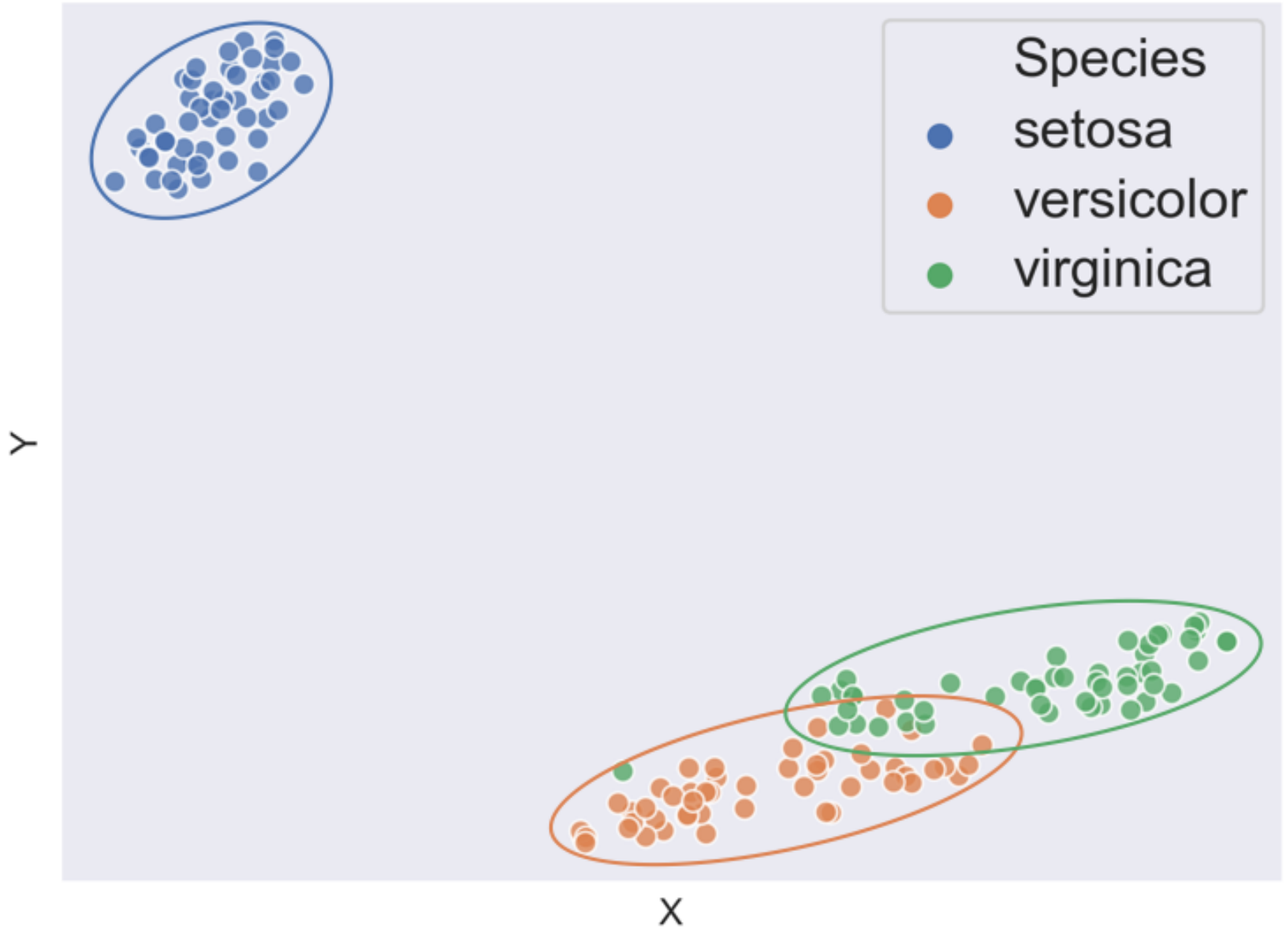
t-SNE on IRIS dataset



t-SNE on IRIS dataset



t-SNE on IRIS dataset



t-SNE on female ANSUR dataset

```
df.shape
```

```
(1986, 99)
```

```
non_numeric = ['BMI_class', 'Height_class',
               'Gender',  'Component', 'Branch']

df_numeric = df.drop(non_numeric, axis=1)

df_numeric.shape
```

```
(1986, 94)
```

Fitting t-SNE

```
from sklearn.manifold import TSNE

m = TSNE(learning_rate=50)
```

```
tsne_features = m.fit_transform(df_numeric)

tsne_features[1:4,:]
```

```
array([[-37.962185,  15.066088],
       [-21.873512,  26.334448],
       [ 13.97476 ,  22.590828]], dtype=float32)
```

Assigning t-SNE features to our dataset

```
tsne_features[1:4,:]
```

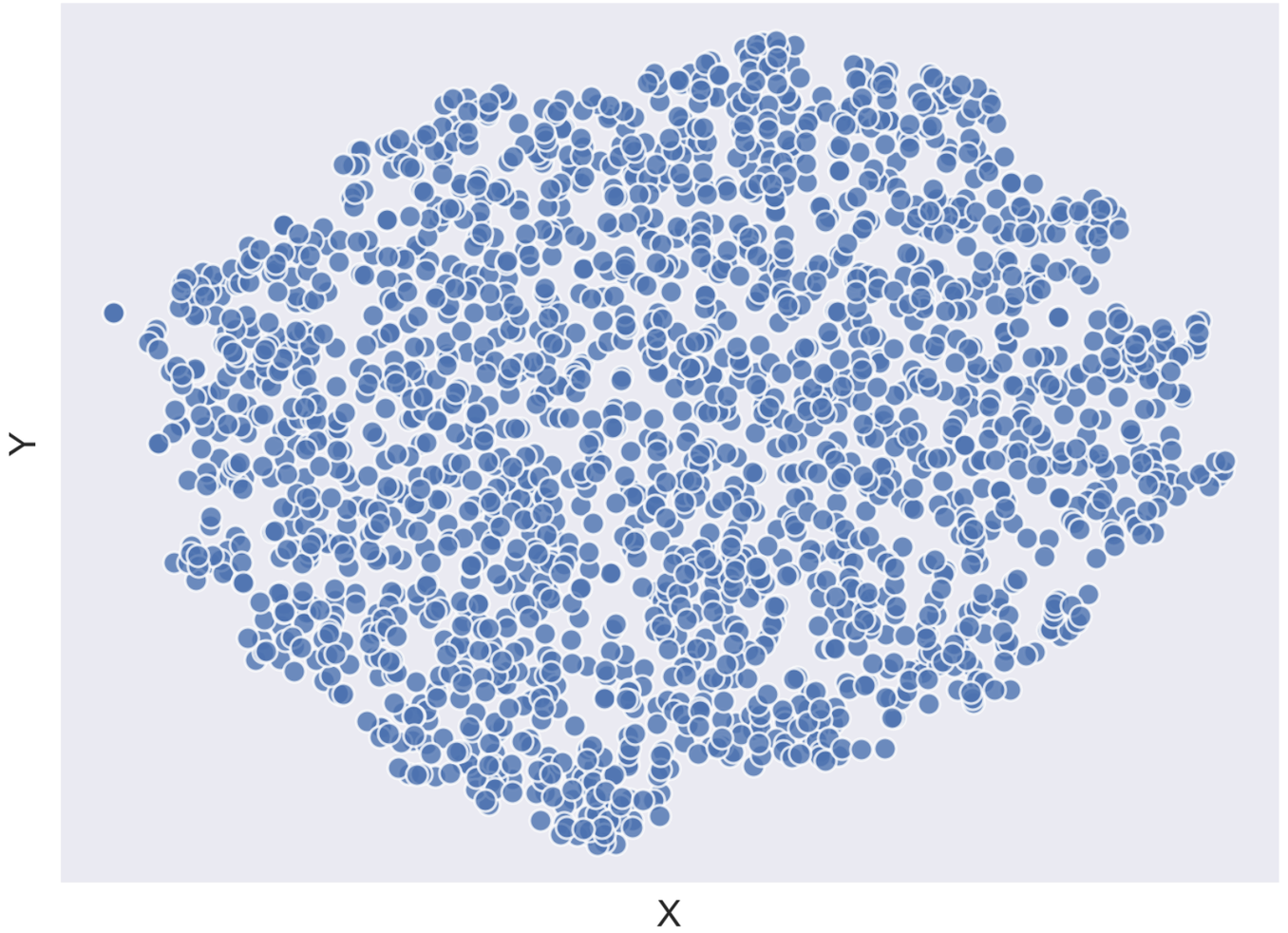
```
array([[-37.962185,  15.066088],
       [-21.873512,  26.334448],
       [ 13.97476 ,  22.590828]], dtype=float32)
```

```
df['x'] = tsne_features[:,0]

df['y'] = tsne_features[:,1]
```

Plotting t-SNE

```
import seaborn as sns

sns.scatterplot(x="x", y="y", data=df)

plt.show()
```

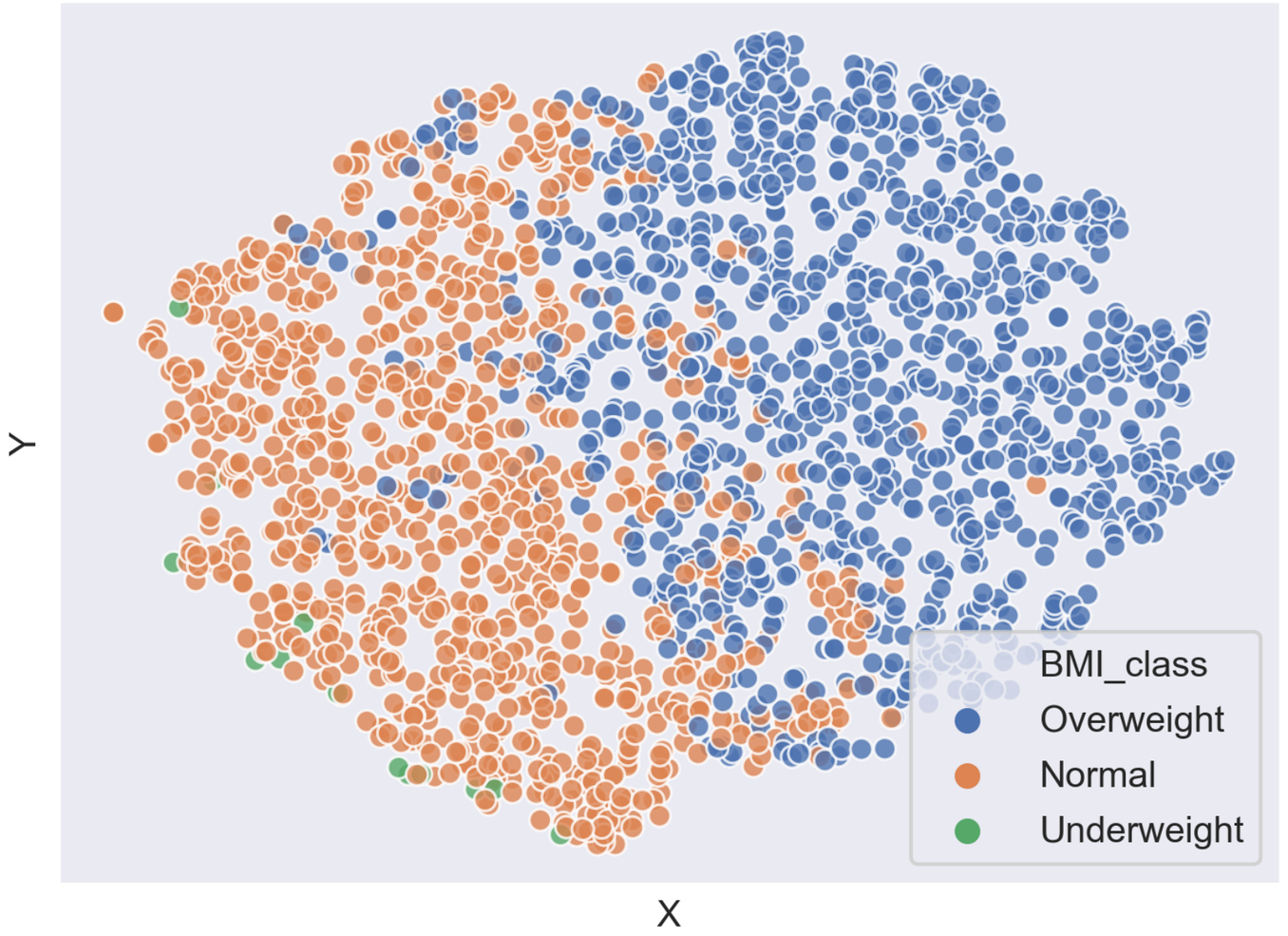
Plotting t-SNE



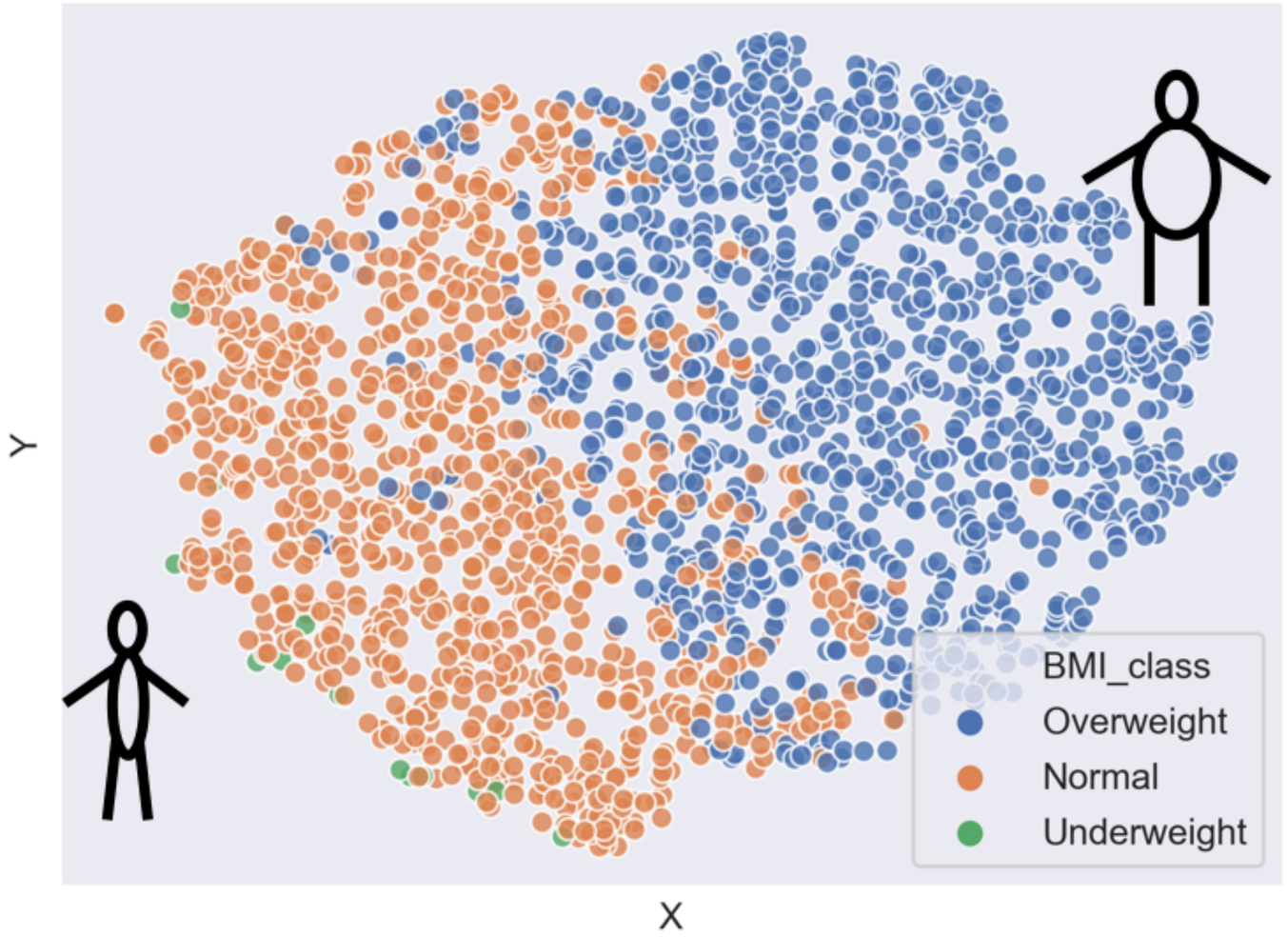
Coloring points according to BMI category

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(x="x", y="y", hue='BMI_class', data=df)

plt.show()
```

Coloring points according to BMI category



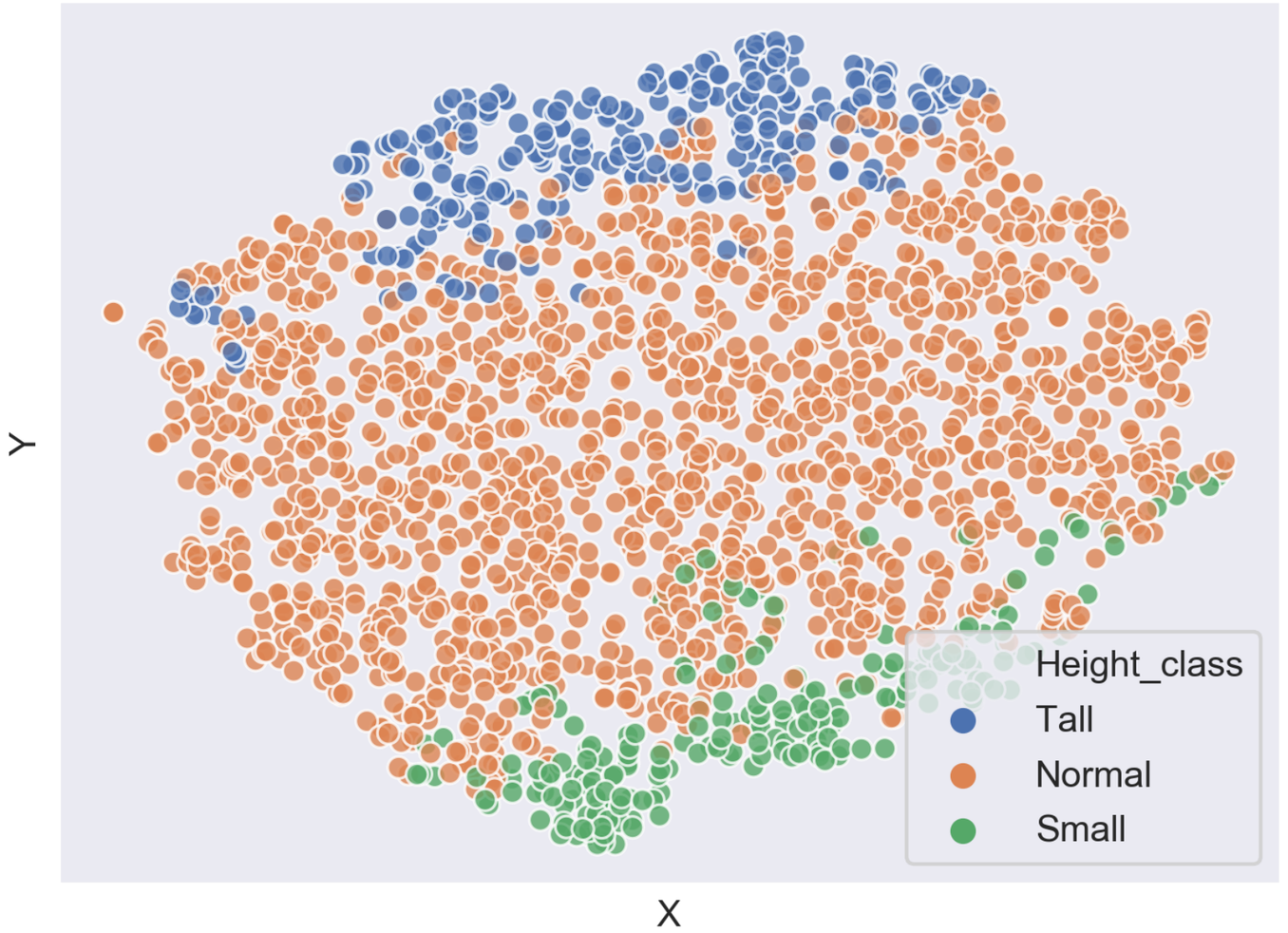
Coloring points according to BMI category



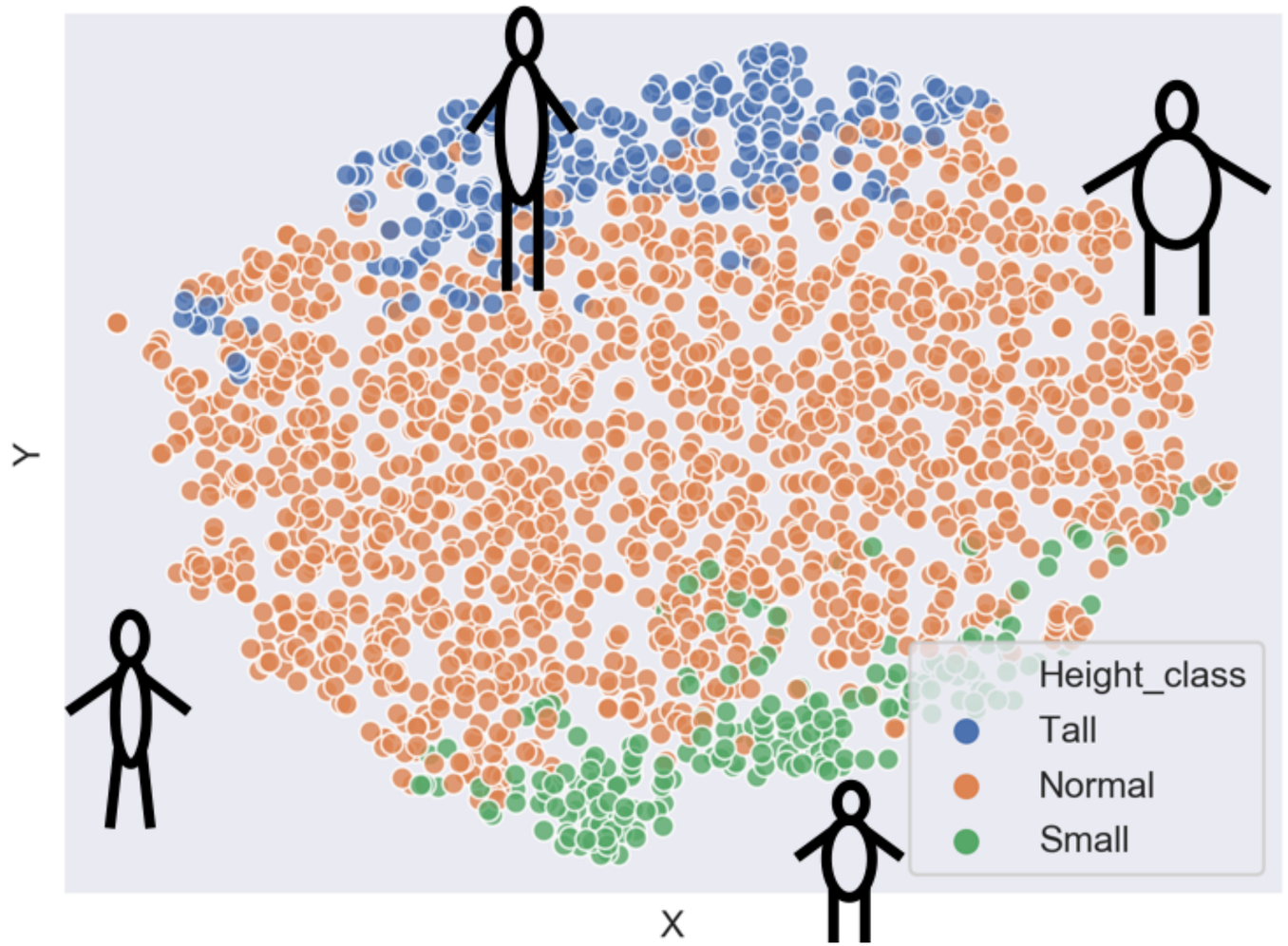
Coloring points according to height category

```
import seaborn as sns

import matplotlib.pyplot as plt

sns.scatterplot(x="x", y="y", hue='Height_class', data=df)

plt.show()
```

Coloring points according to height category



Coloring points according to height category



Let's practice!

DIMENSIONALITY REDUCTION IN PYTHON