Machine Learning

A Beginner's Practical Guide

Michael Moussa

@michaelmoussa

https://joind.in/talk/5f55e

What is "Machine Learning"?

"Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead."

Arthur Samuel (1901-1990) playing checkers on the IBM 701

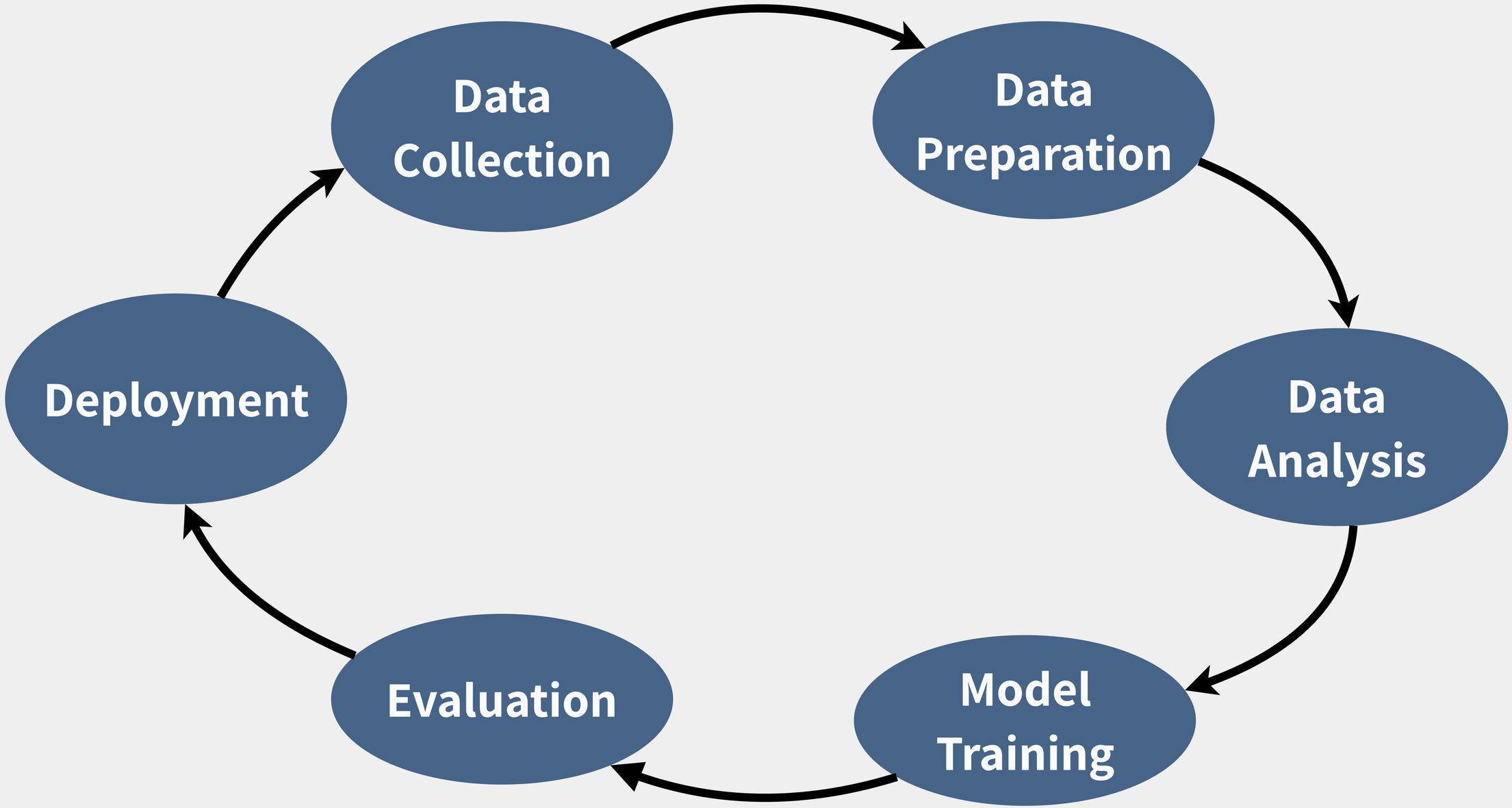
https://en.wikipedia.org/wiki/Machine_learning

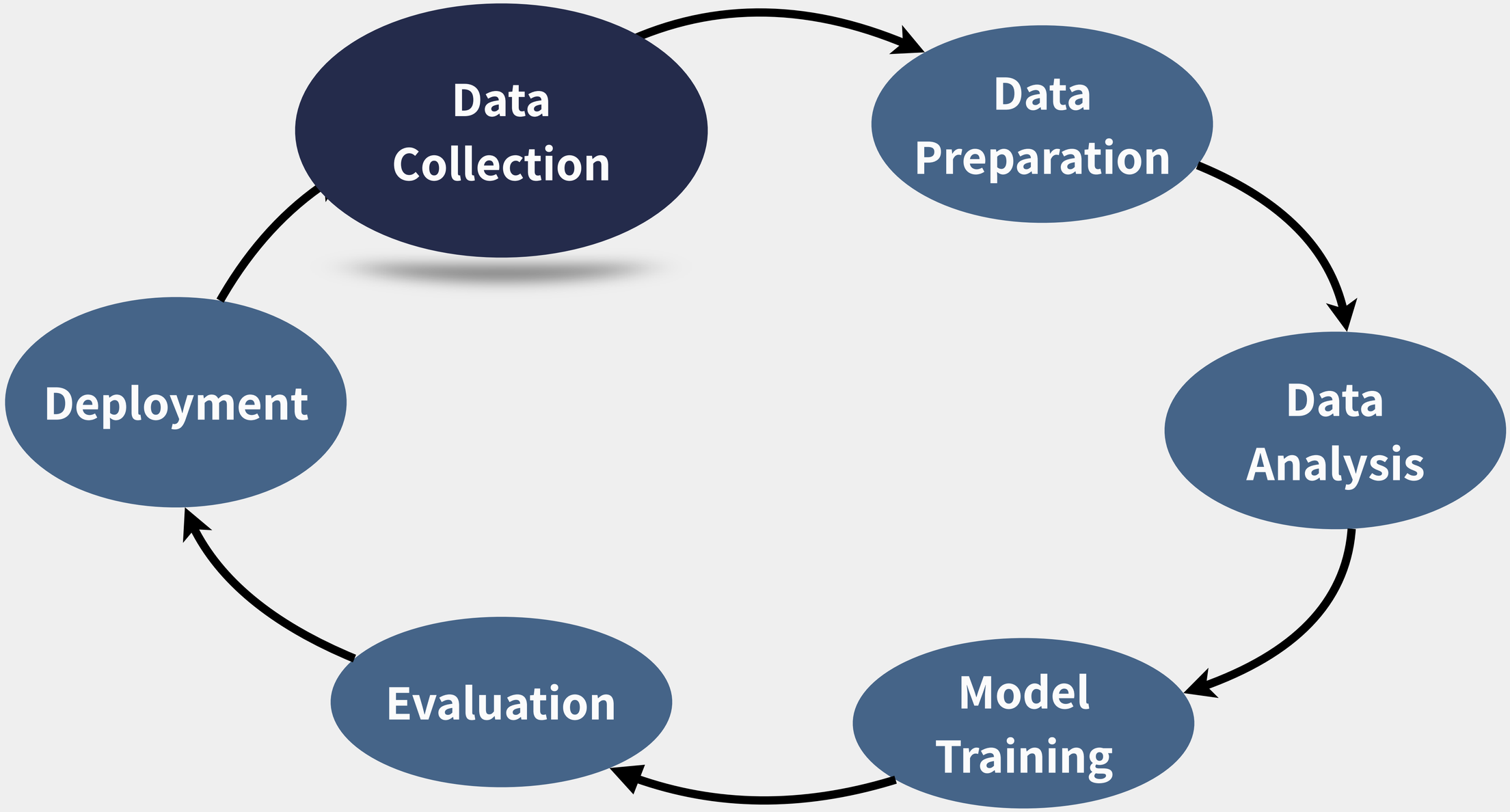
What is "Machine Learning"?

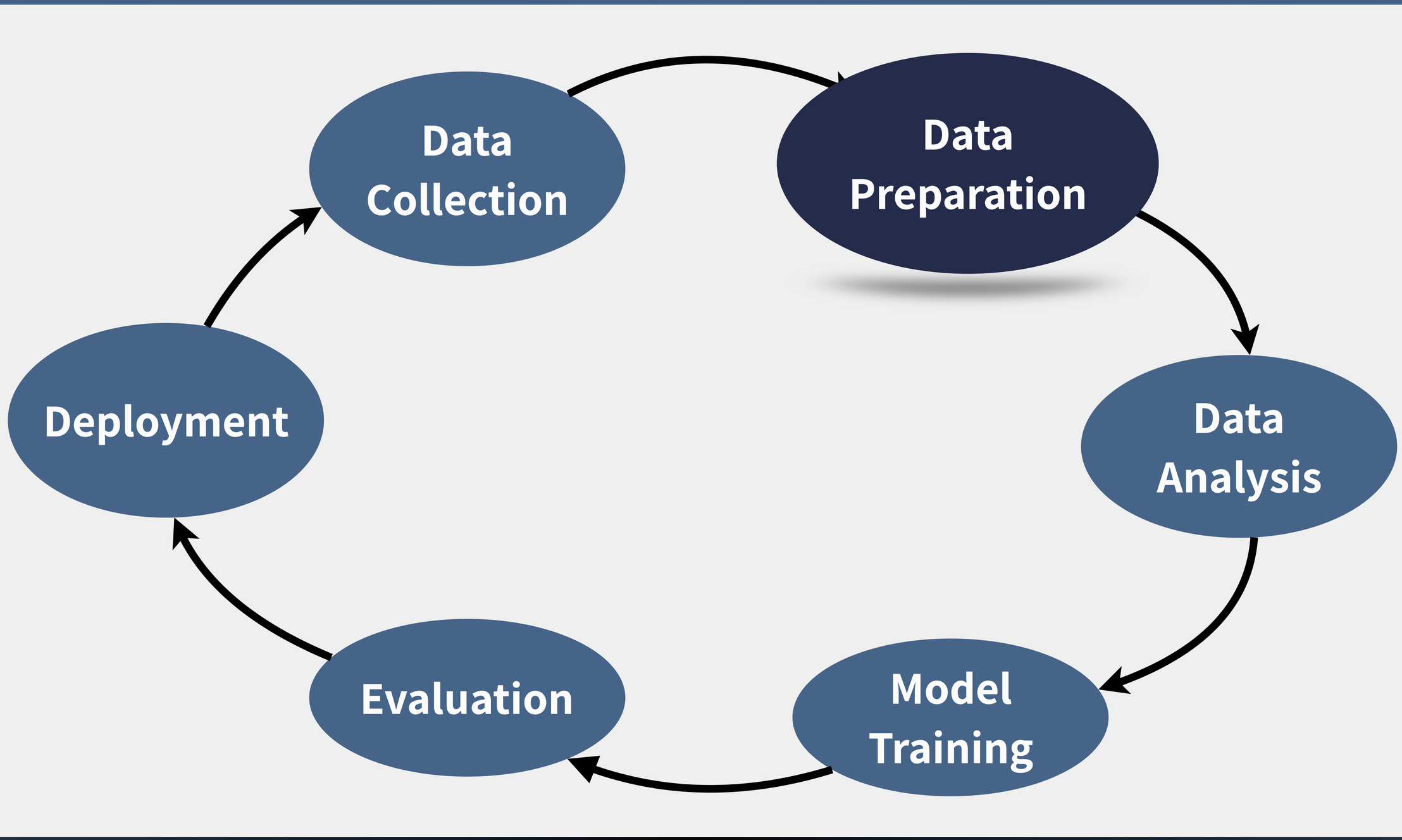
"Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead."

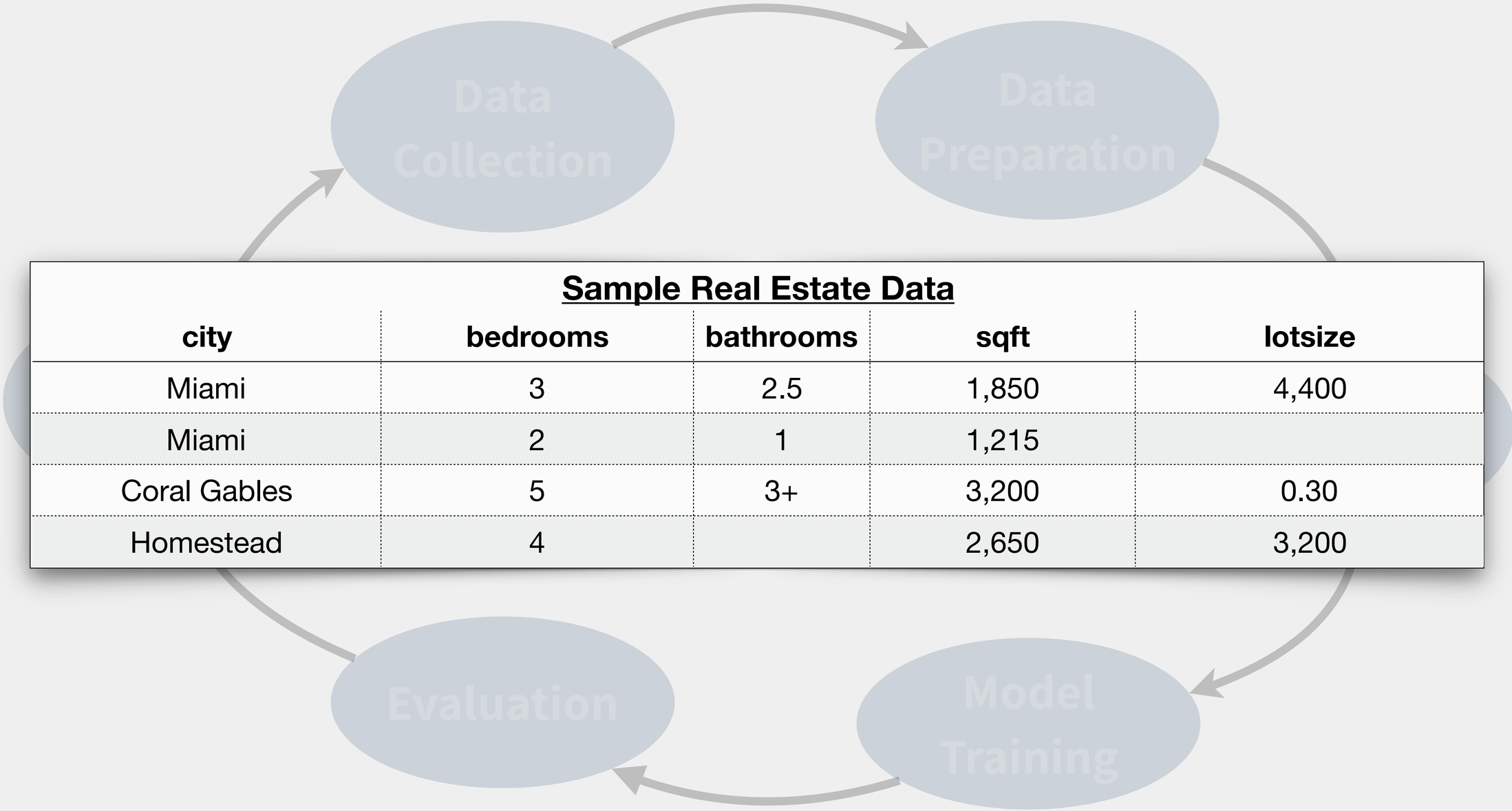
Arthur Samuel (1901-1990) playing checkers on the IBM 701

https://en.wikipedia.org/wiki/Machine_learning


Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment

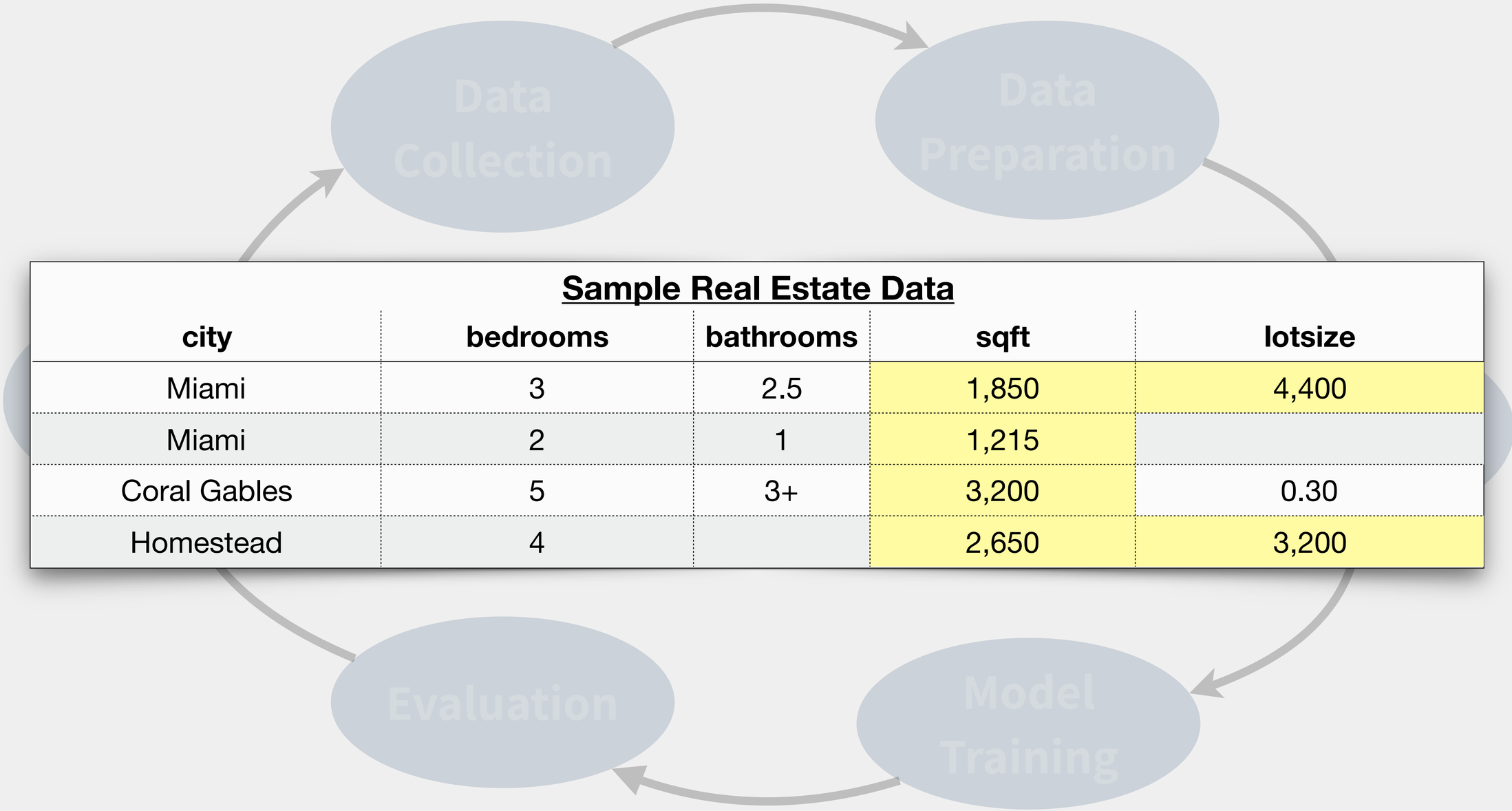

Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment


Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment



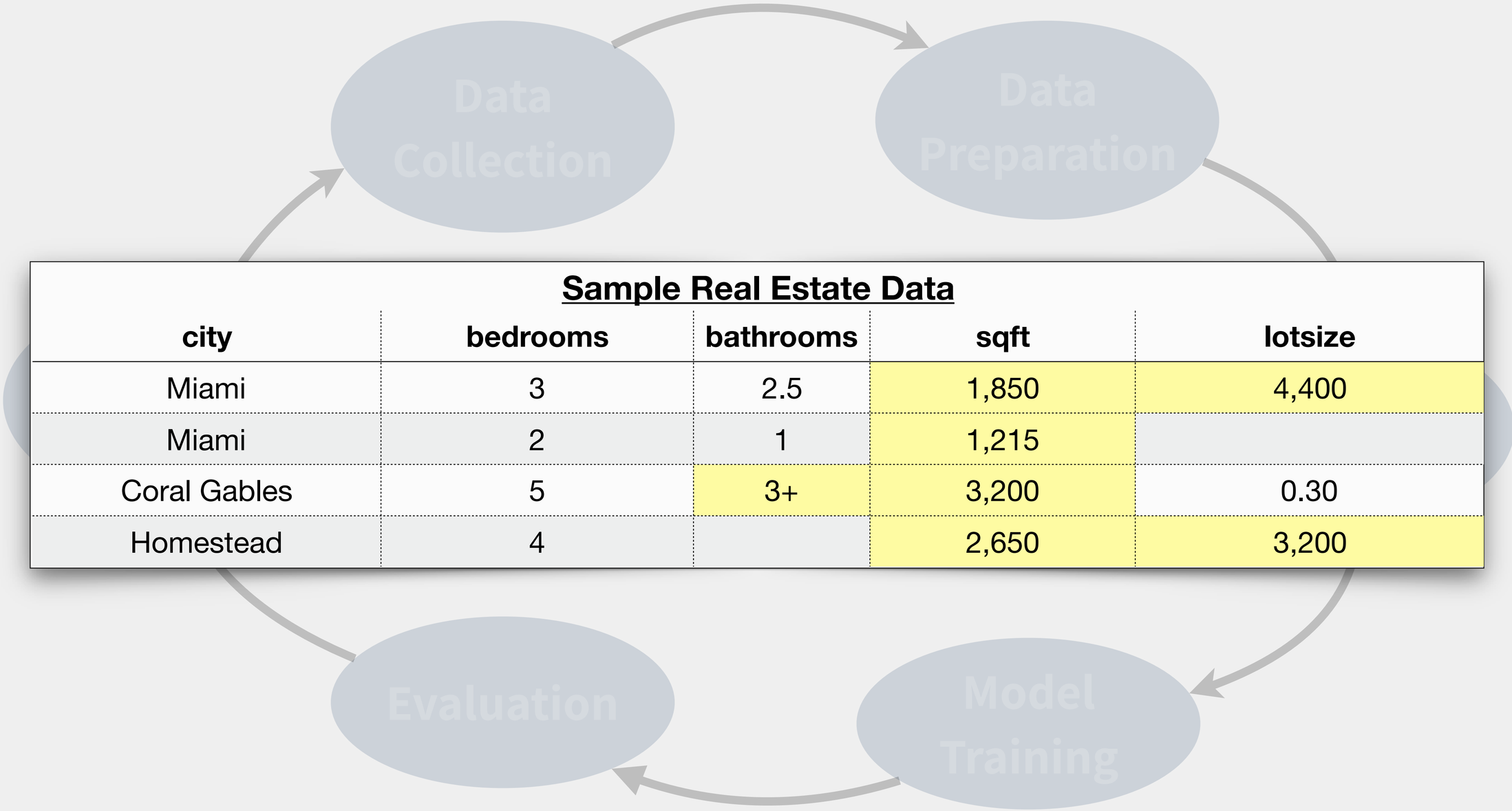
Sample Real Estate Data

city	bedrooms	bathrooms	sqft	lotsize
Miami	3	2.5	1,850	4,400
Miami	2	1	1,215	
Coral Gables	5	3+	3,200	0.30
Homestead	4		2,650	3,200



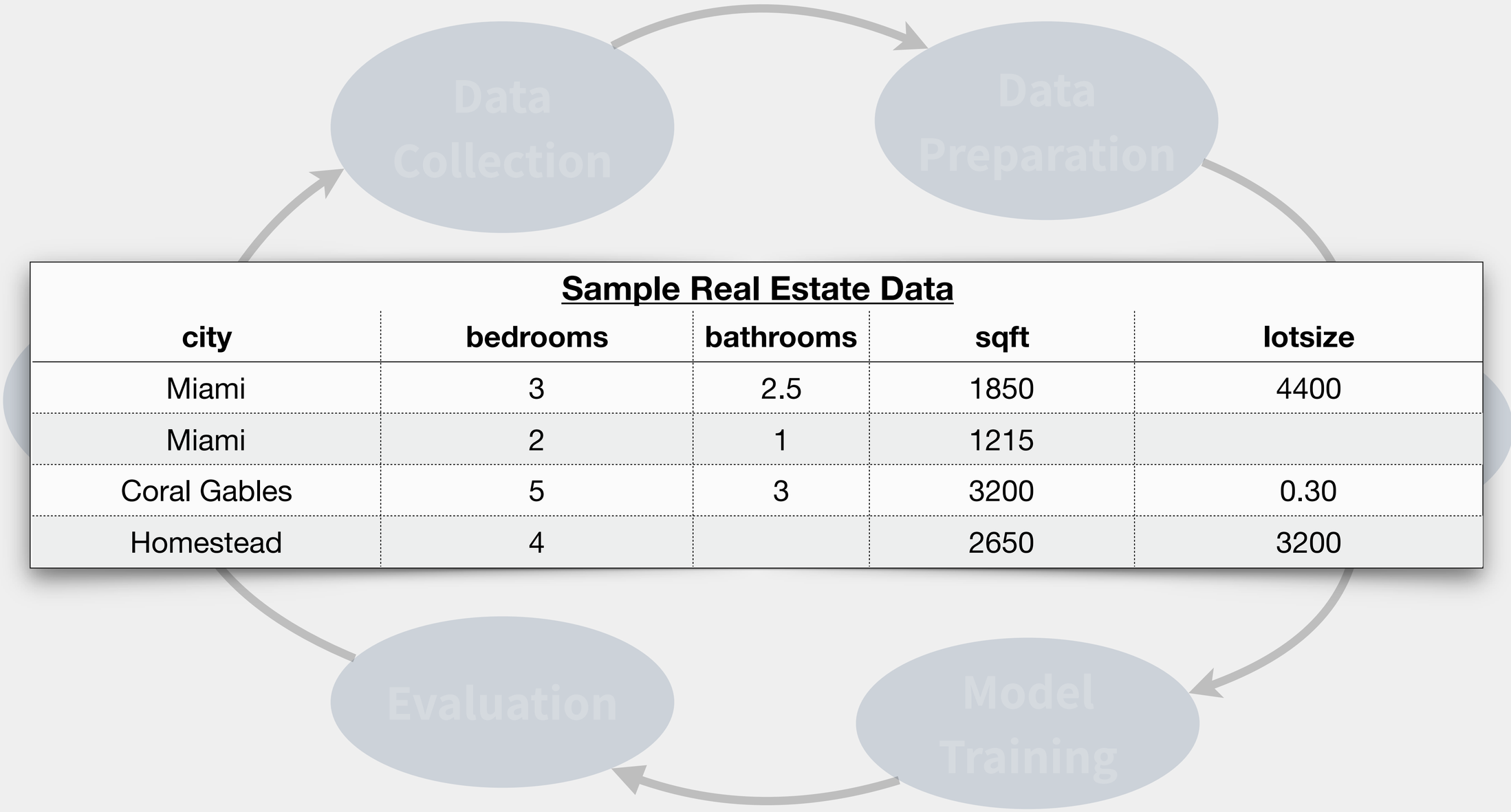
Sample Real Estate Data

city	bedrooms	bathrooms	sqft	lotsize
Miami	3	2.5	1,850	4,400
Miami	2	1	1,215	
Coral Gables	5	3+	3,200	0.30
Homestead	4		2,650	3,200



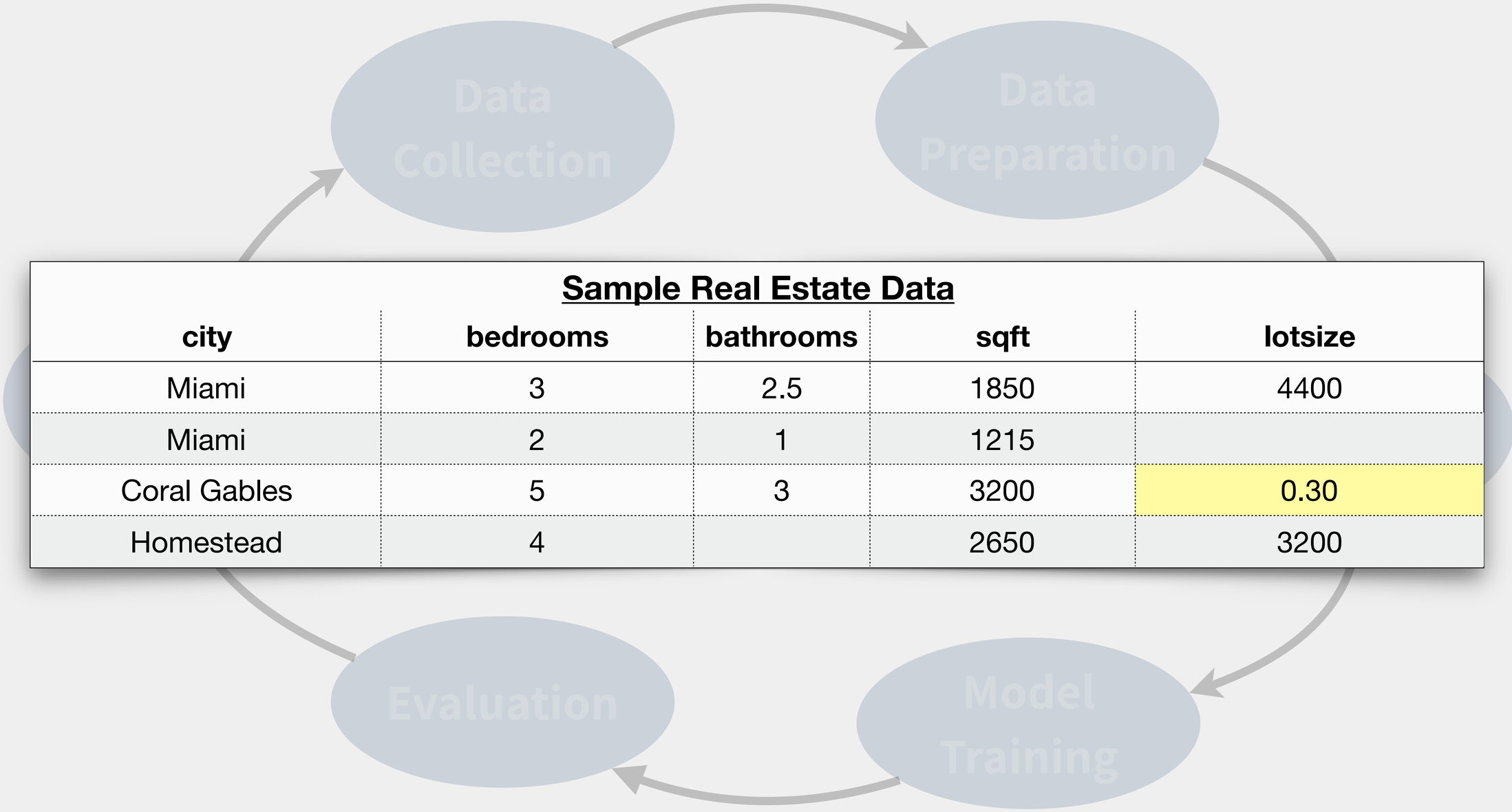
Sample Real Estate Data

city	bedrooms	bathrooms	sqft	lotsize
Miami	3	2.5	1,850	4,400
Miami	2	1	1,215	
Coral Gables	5	3+	3,200	0.30
Homestead	4		2,650	3,200



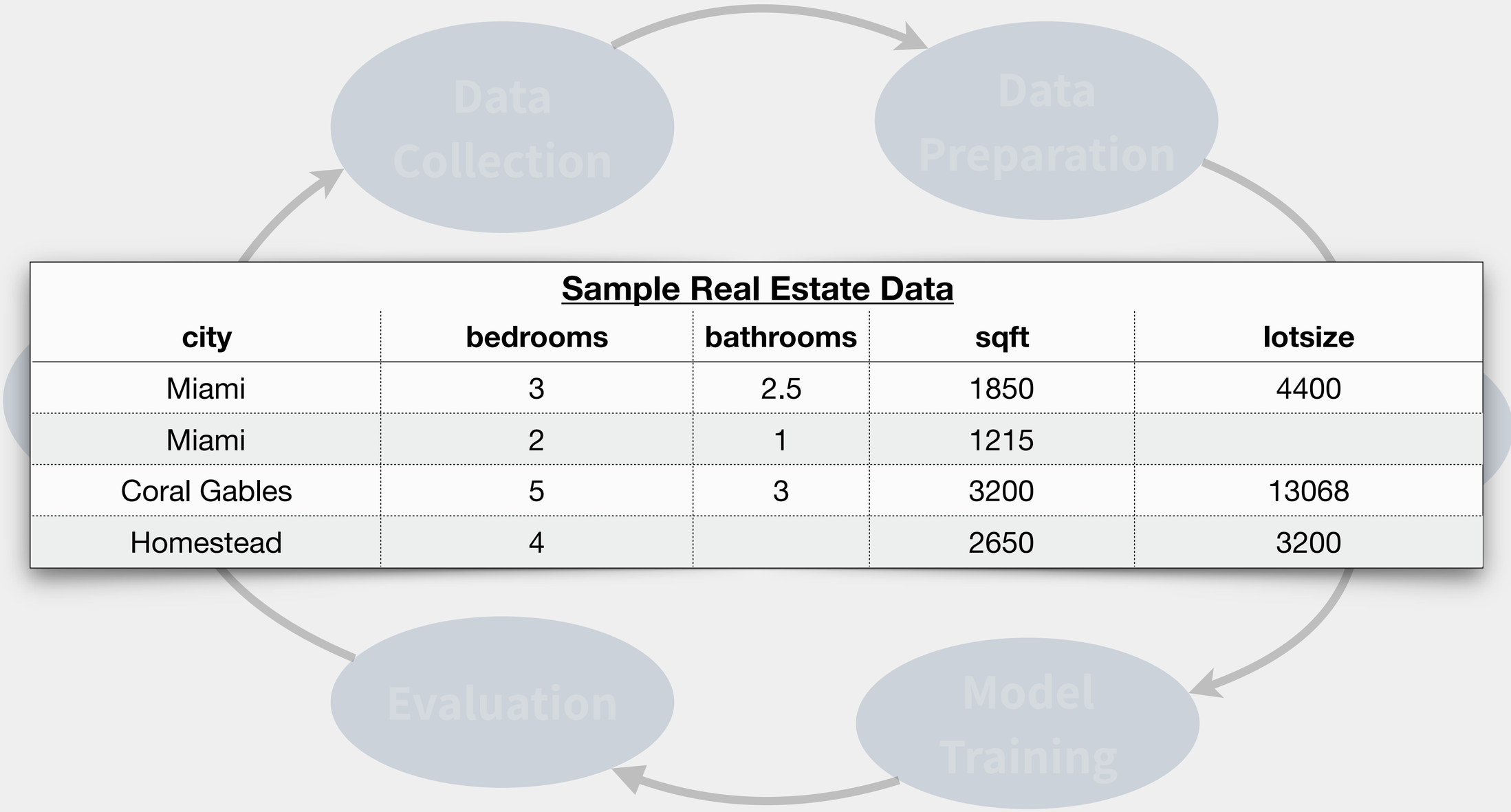
Sample Real Estate Data

city	bedrooms	bathrooms	sqft	lotsize
Miami	3	2.5	1850	4400
Miami	2	1	1215	
Coral Gables	5	3	3200	0.30
Homestead	4		2650	3200



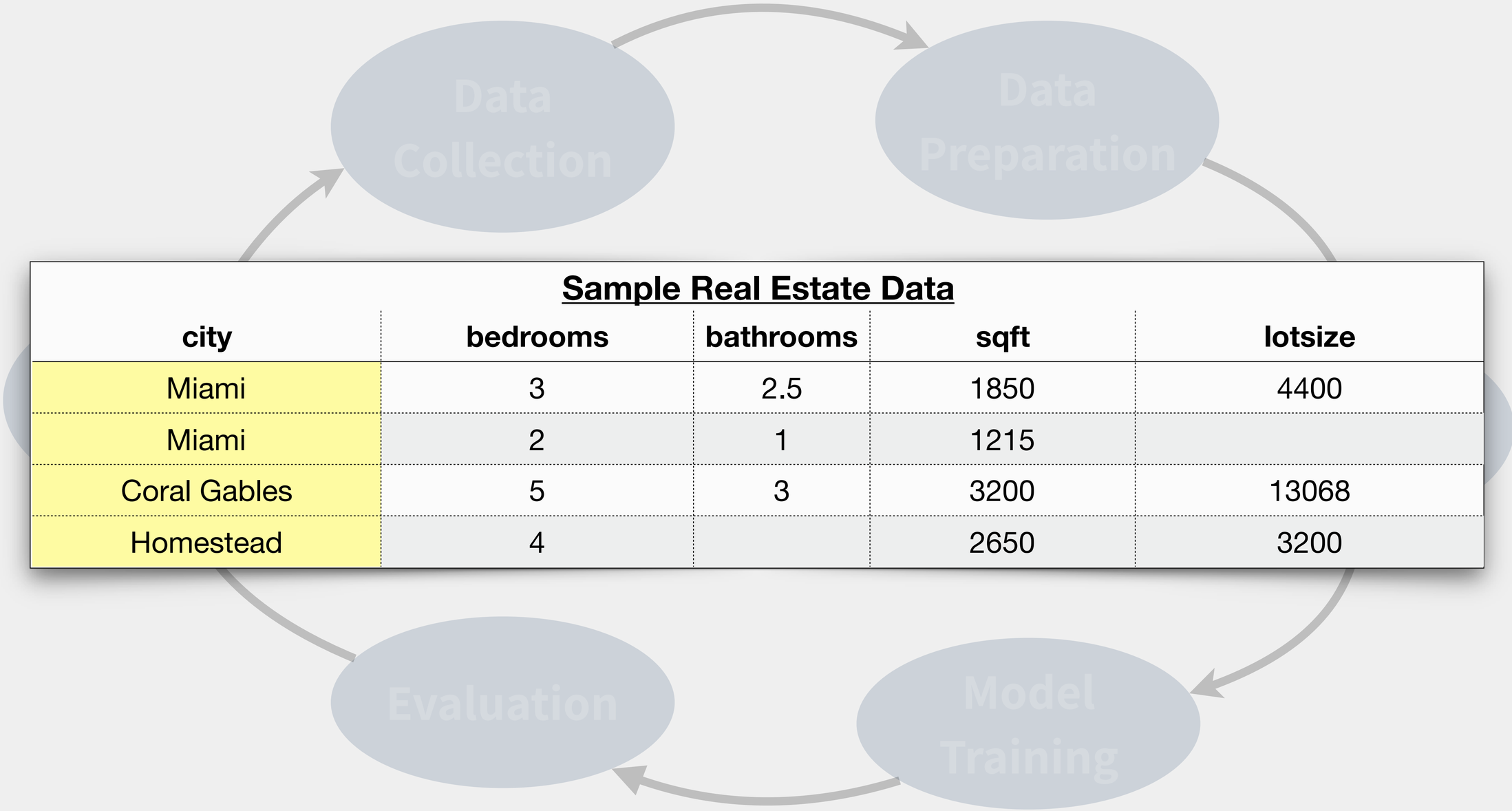
Sample Real Estate Data

city	bedrooms	bathrooms	sqft	lotsize
Miami	3	2.5	1850	4400
Miami	2	1	1215	
Coral Gables	5	3	3200	0.30
Homestead	4		2650	3200



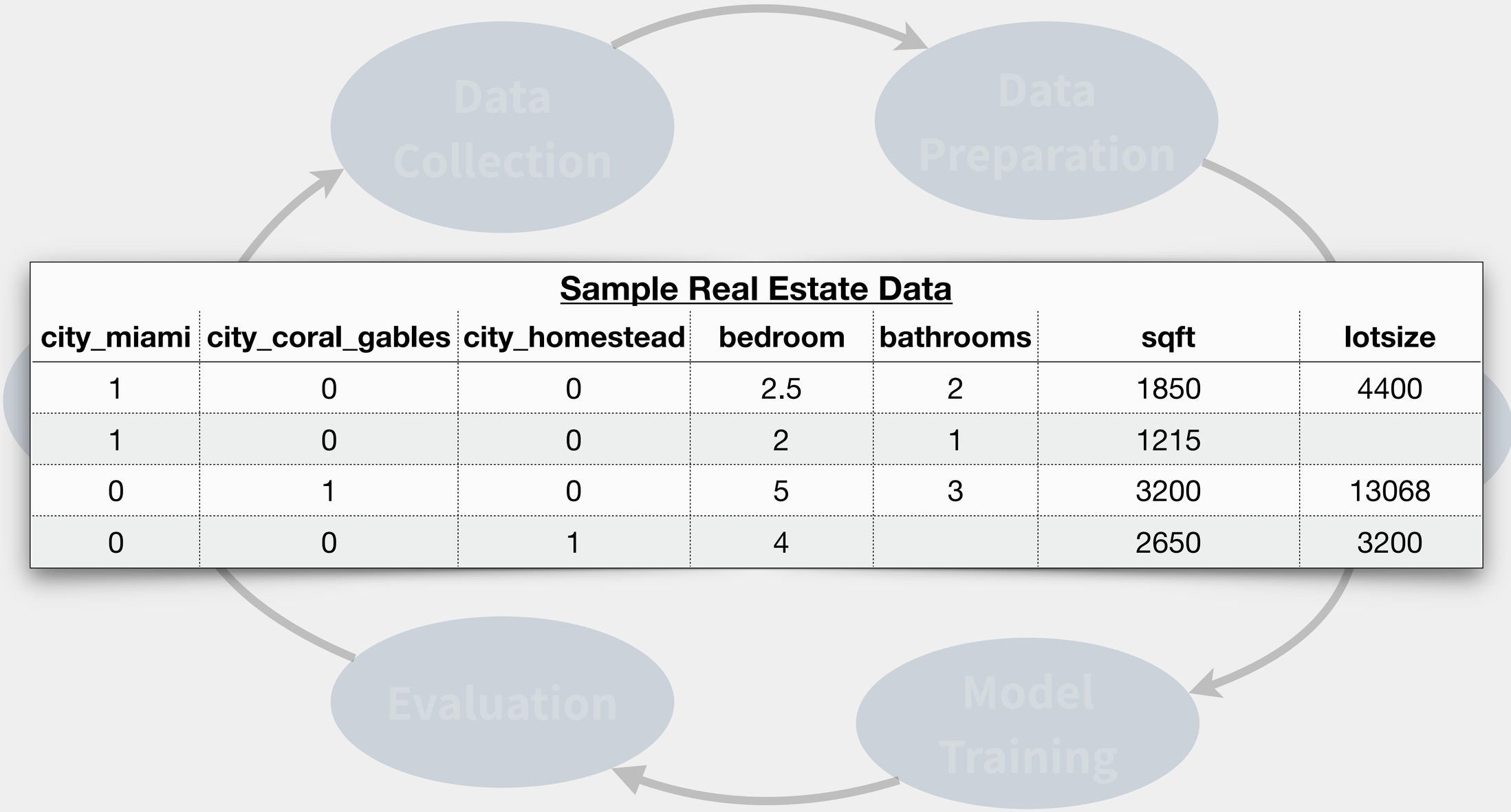
Sample Real Estate Data

city	bedrooms	bathrooms	sqft	lotsize
Miami	3	2.5	1850	4400
Miami	2	1	1215	
Coral Gables	5	3	3200	13068
Homestead	4		2650	3200



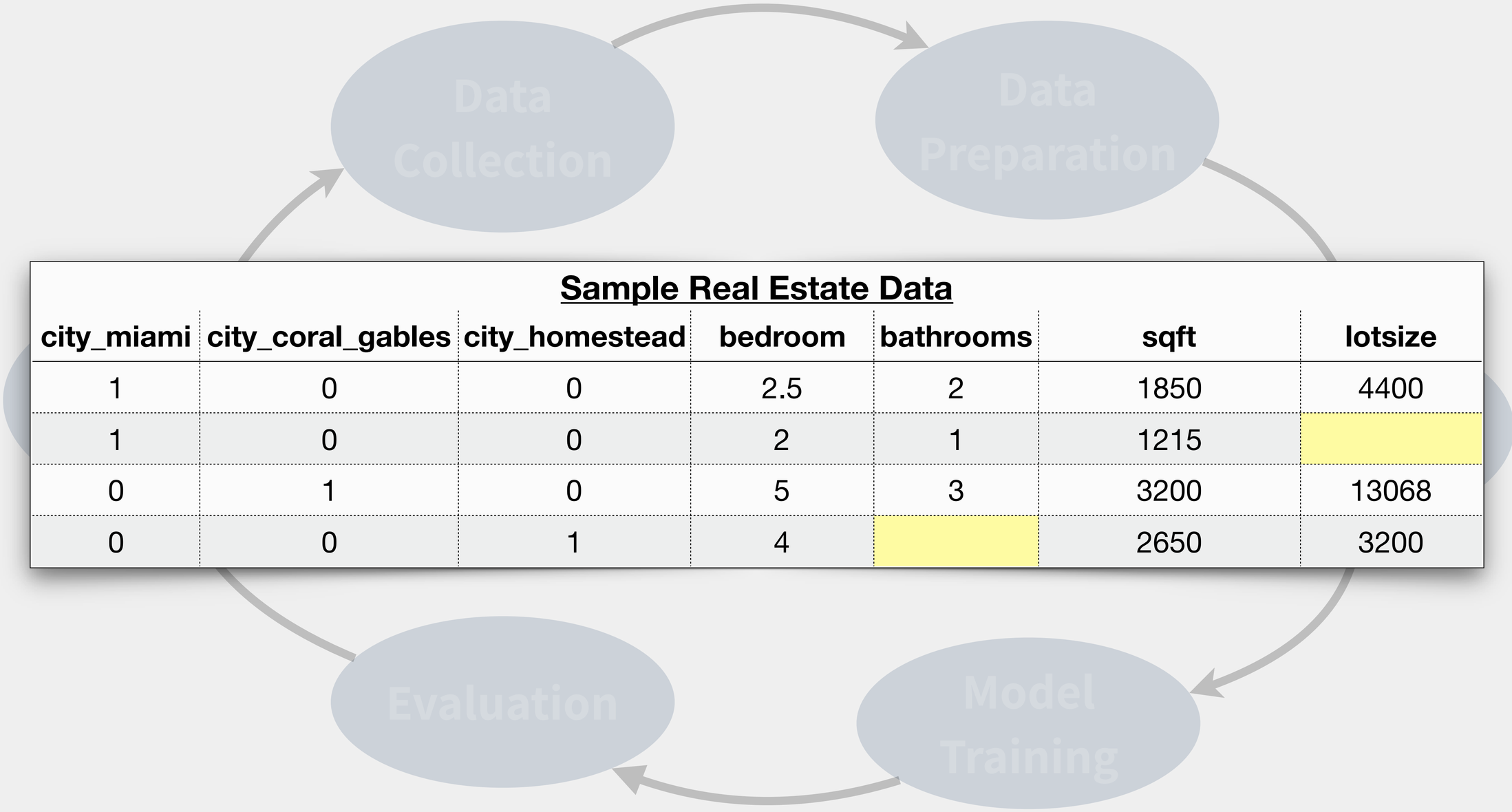
Sample Real Estate Data

city	bedrooms	bathrooms	sqft	lotsize
Miami	3	2.5	1850	4400
Miami	2	1	1215	
Coral Gables	5	3	3200	13068
Homestead	4		2650	3200



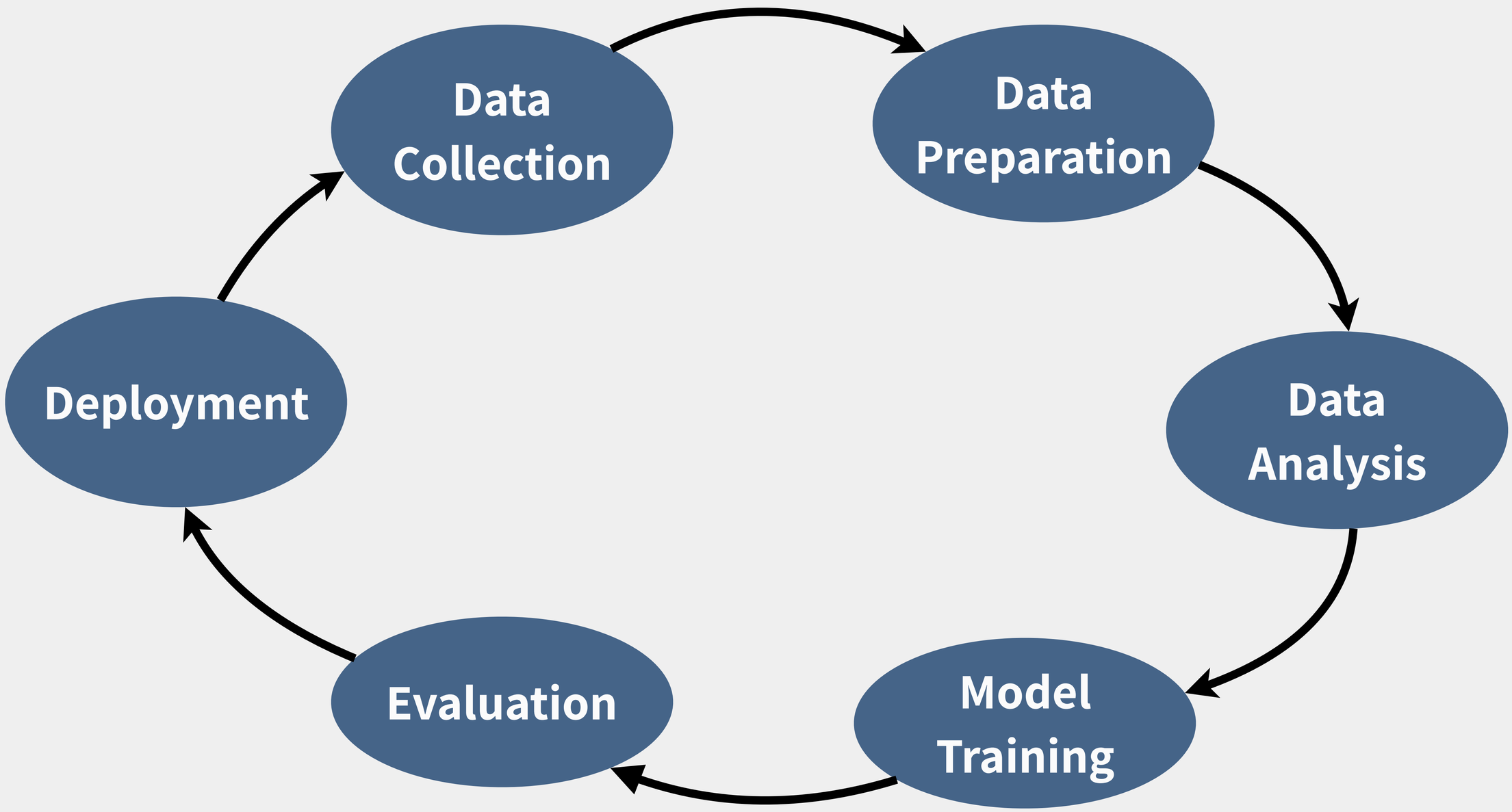
Sample Real Estate Data

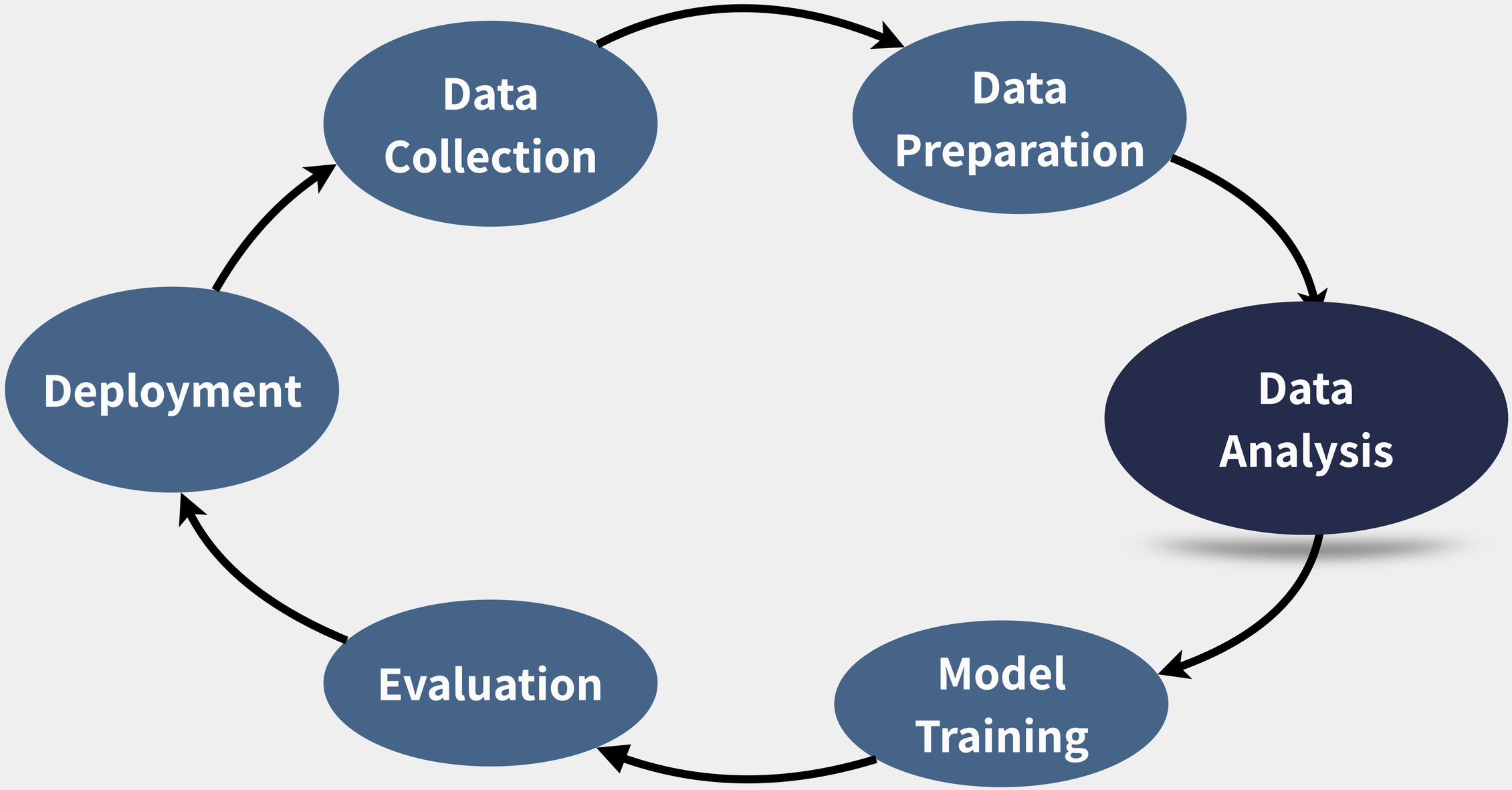
city_miami	city_coral_gables	city_homestead	bedroom	bathrooms	sqft	lotsize
1	0	0	2.5	2	1850	4400
1	0	0	2	1	1215	
0	1	0	5	3	3200	13068
0	0	1	4		2650	3200



Sample Real Estate Data

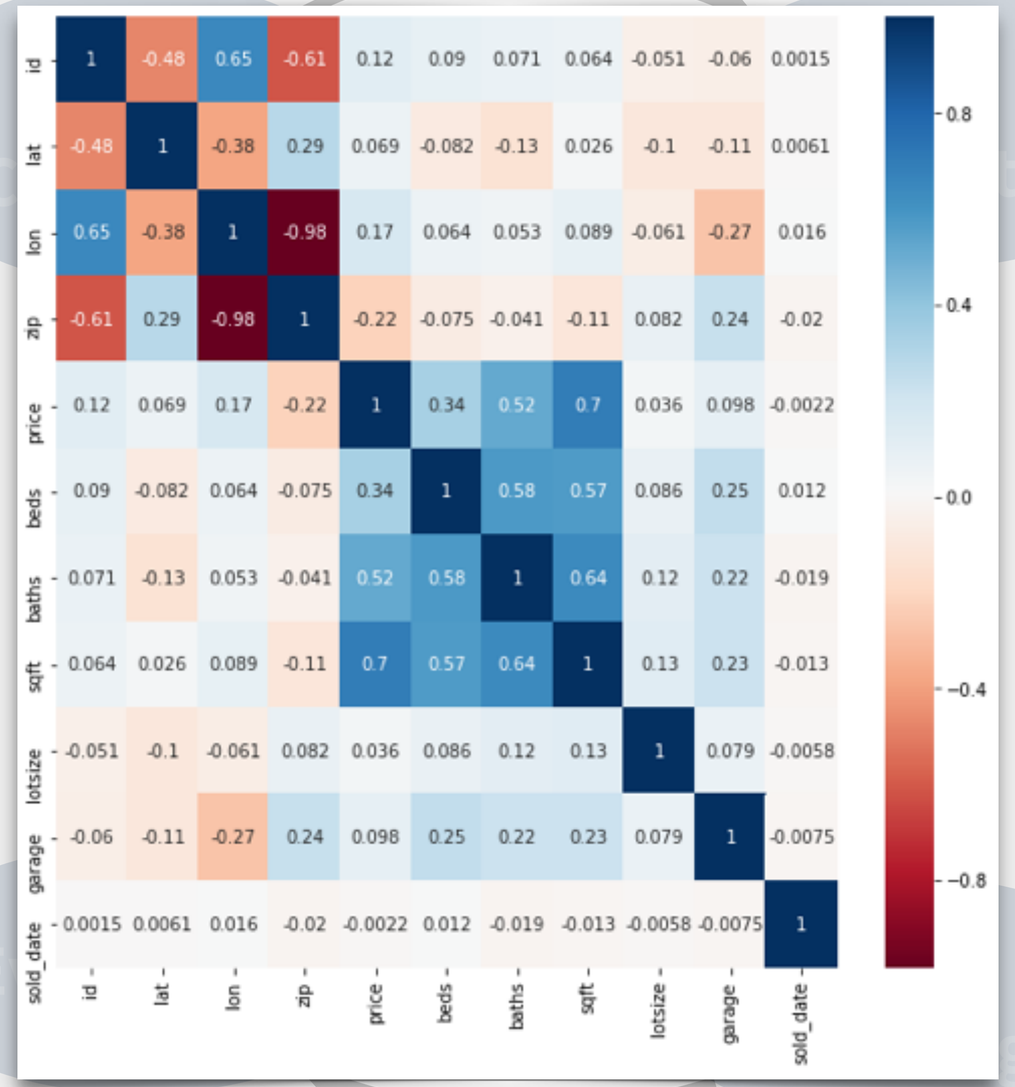
city_miami	city_coral_gables	city_homestead	bedroom	bathrooms	sqft	lotsize
1	0	0	2.5	2	1850	4400
1	0	0	2	1	1215	
0	1	0	5	3	3200	13068
0	0	1	4		2650	3200


Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment


Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment

Deployment

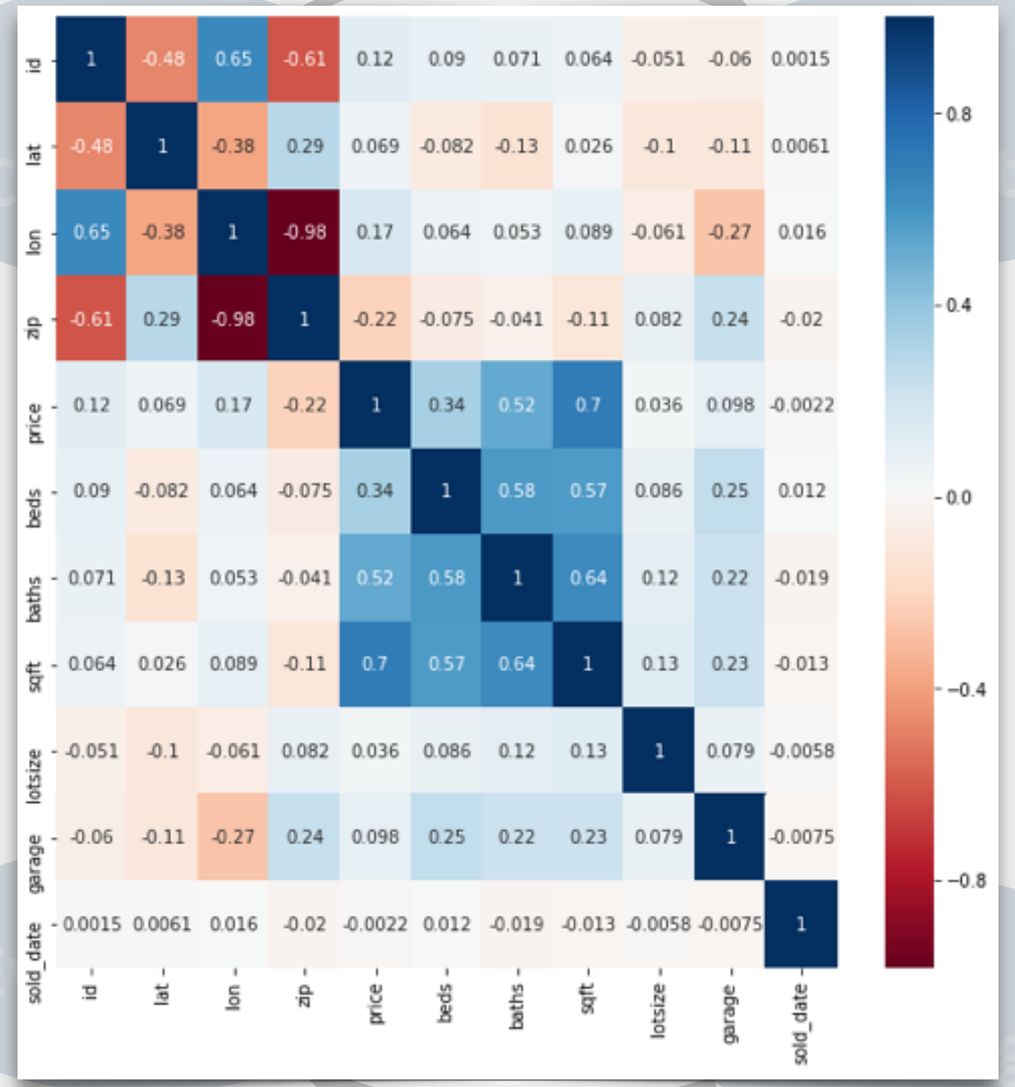
Data Analysis

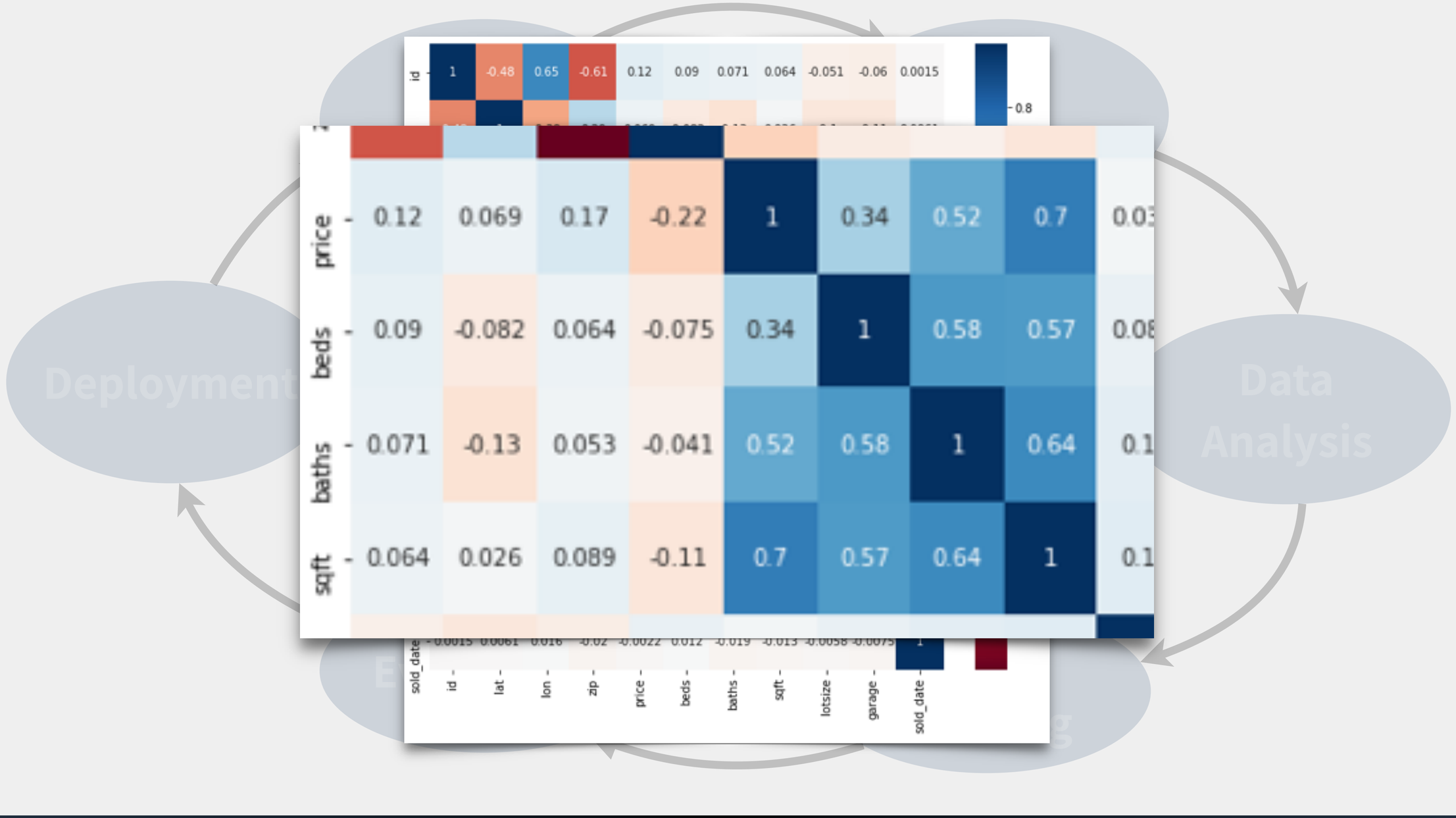


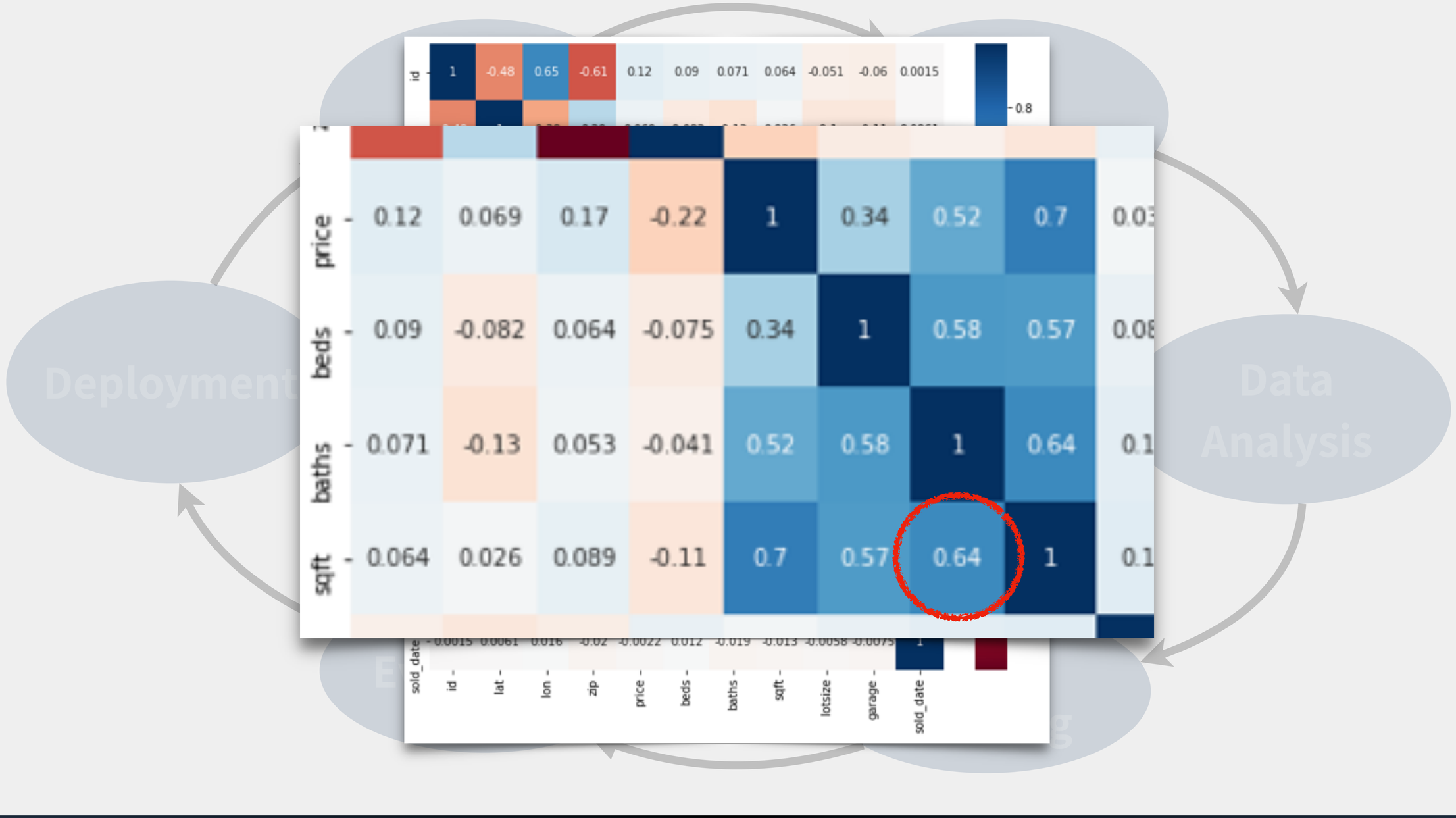
id
lat
lon
zip
price
beds
baths
sqft
lotsize
garage
sold_date
1 -0.48 0.65 -0.61 0.12 0.09 0.071 0.064 -0.051 -0.06 0.0015
-0.48 1 -0.38 0.29 0.069 -0.082 -0.13 0.026 -0.1 -0.11 0.0061
0.65 -0.38 1 -0.98 0.17 0.064 0.053 0.089 -0.061 -0.27 0.016
-0.61 0.29 -0.98 1 -0.22 -0.075 -0.041 -0.11 0.082 0.24 -0.02
0.8
0.4
-0.8

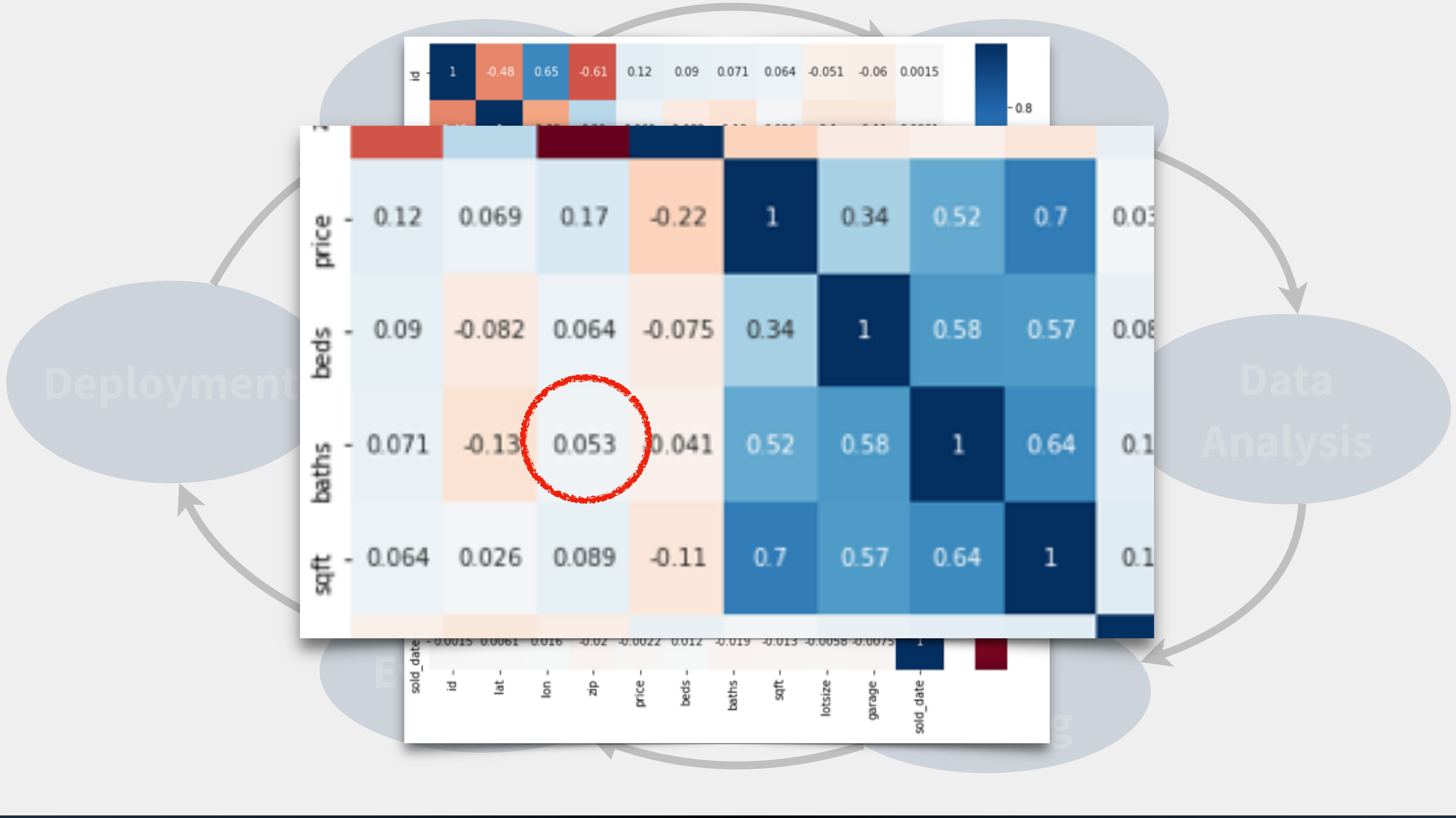
Deployment

Data Analysis



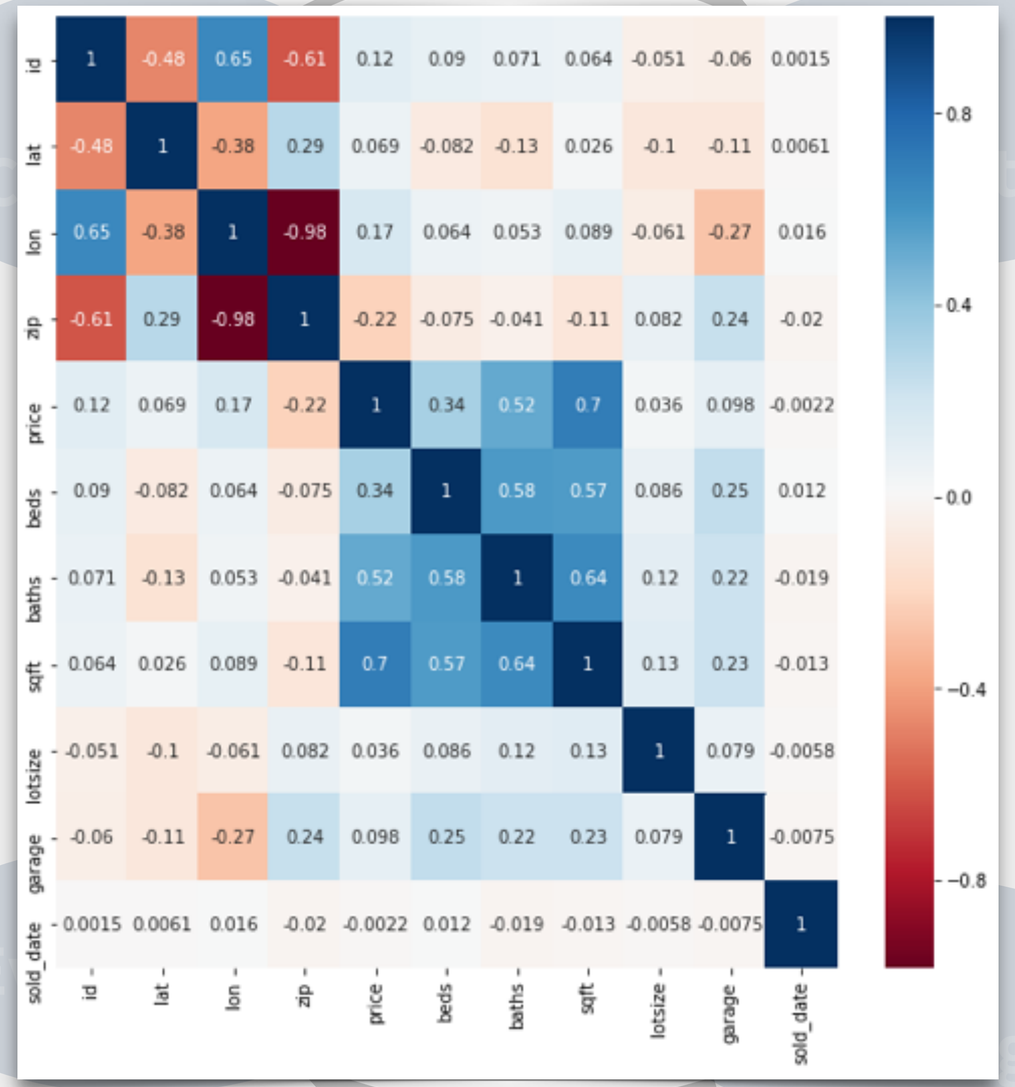

Deployment
Data Analysis
id
price
beds
baths
sqft
sold_date
lat
lon
zip
lotsize
garage
0.8


Deployment
Data Analysis
price
beds
baths
sqft
0.12 0.069 0.17 -0.22 1 0.34 0.52 0.7
0.09 -0.082 0.064 -0.075 0.34 1 0.58 0.57
0.071 -0.13 0.053 -0.041 0.52 0.58 1 0.64
0.064 0.026 0.089 -0.11 0.7 0.57 0.64 1
id
lat
lon
zip
price
beds
baths
sqft
lotsize
garage
sold_date


Deployment
Data Analysis
id
price
beds
baths
sqft
sold_date
lat
lon
zip
lotsize
garage
0.8

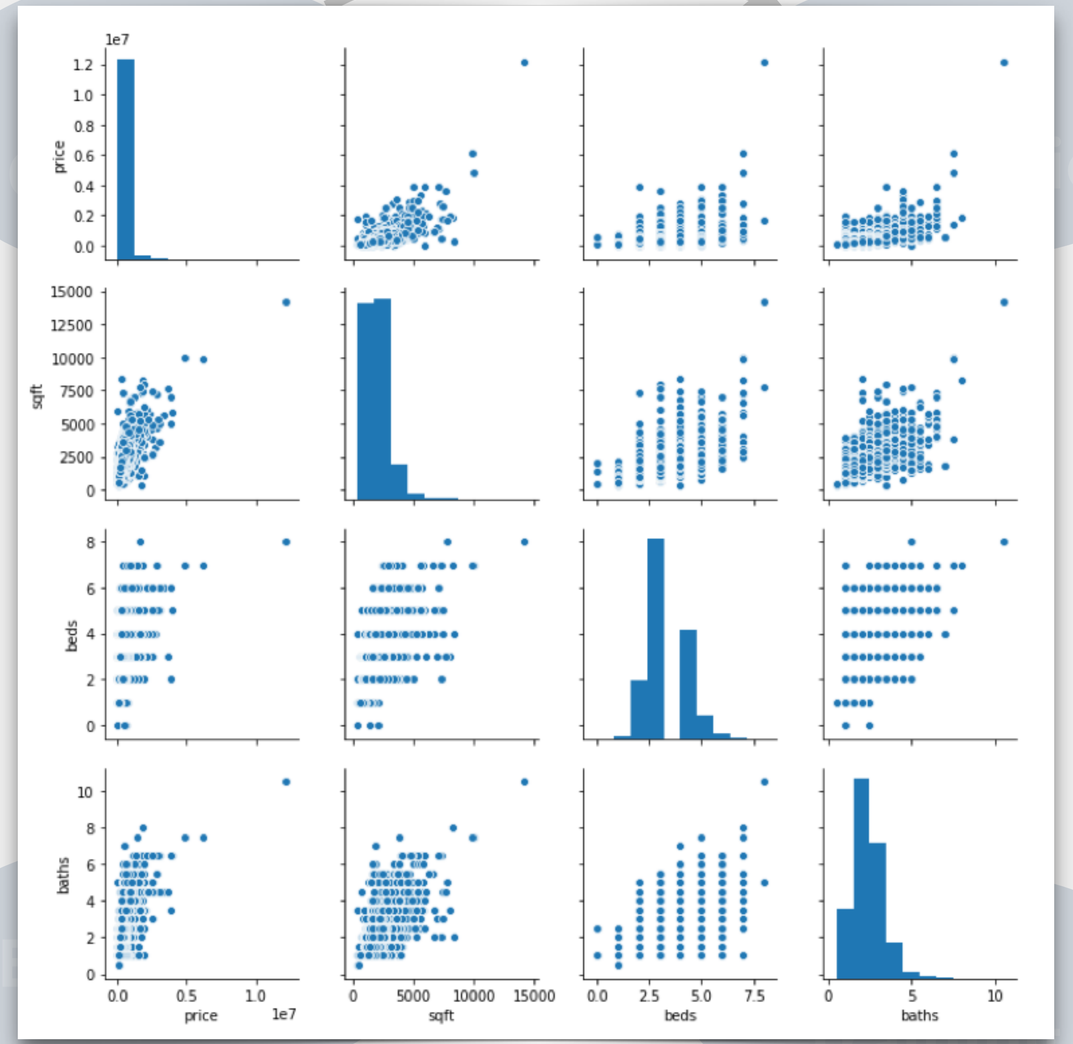
Deployment

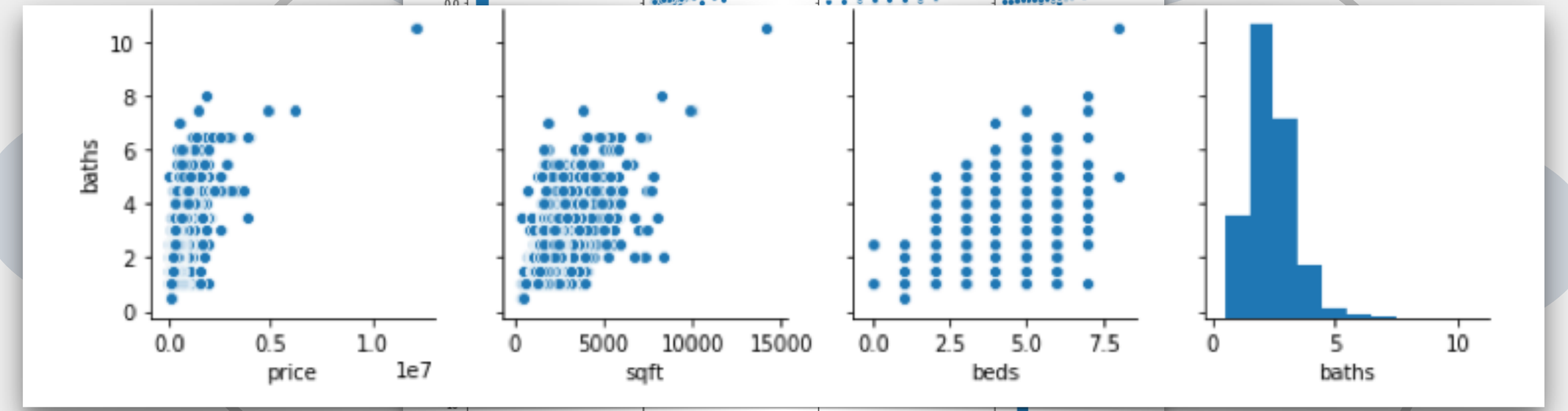
Data Analysis



Deployment

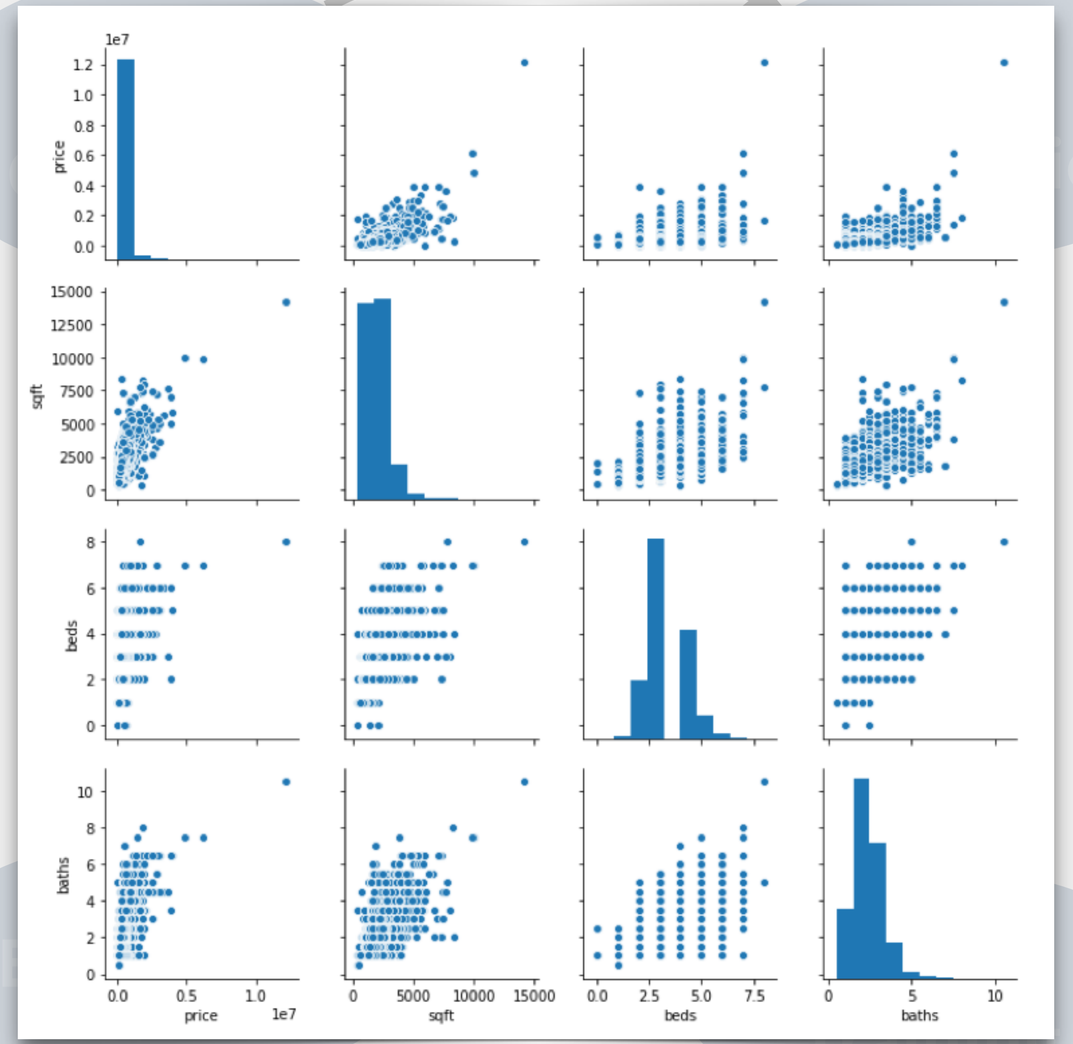
Data Analysis

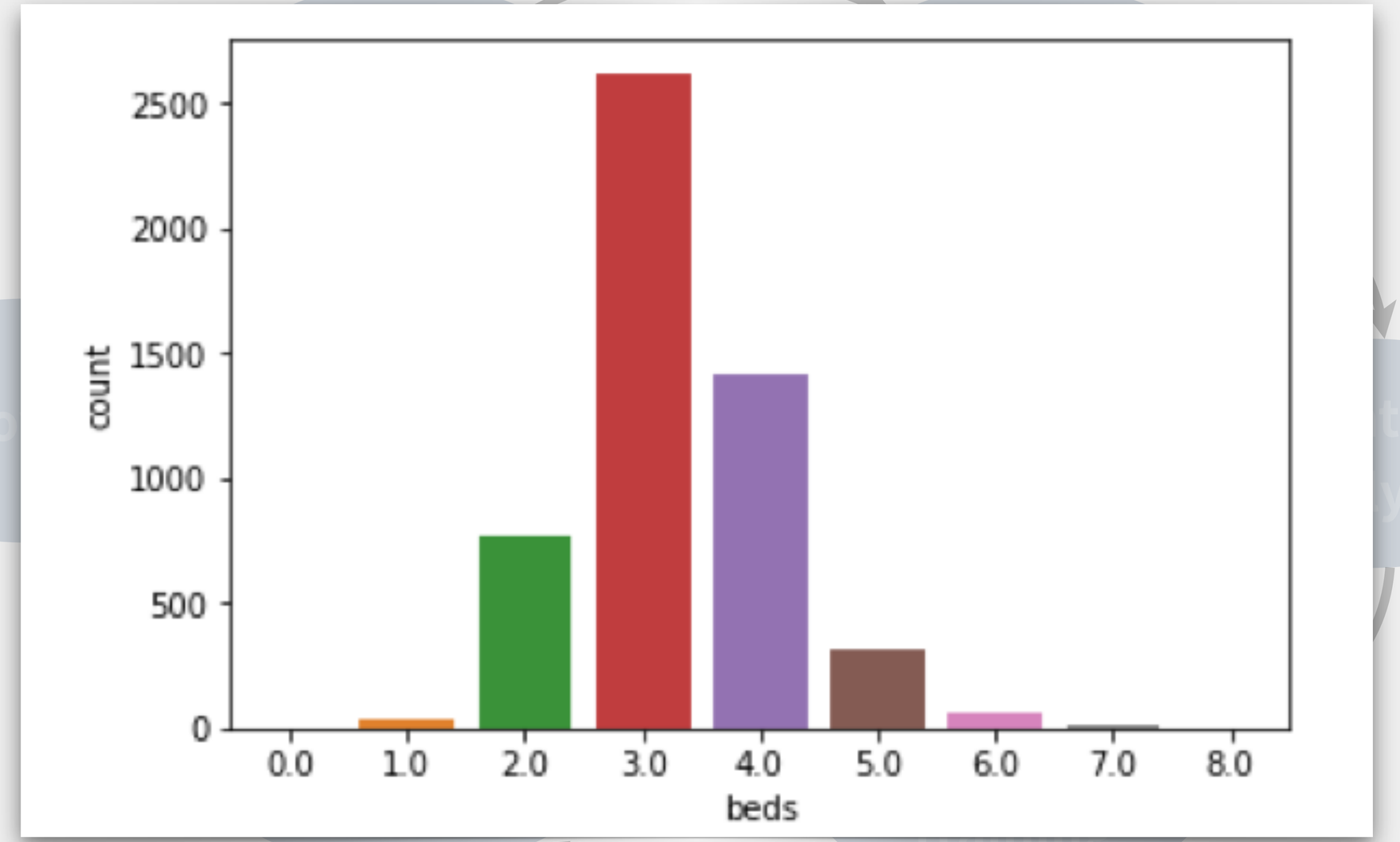


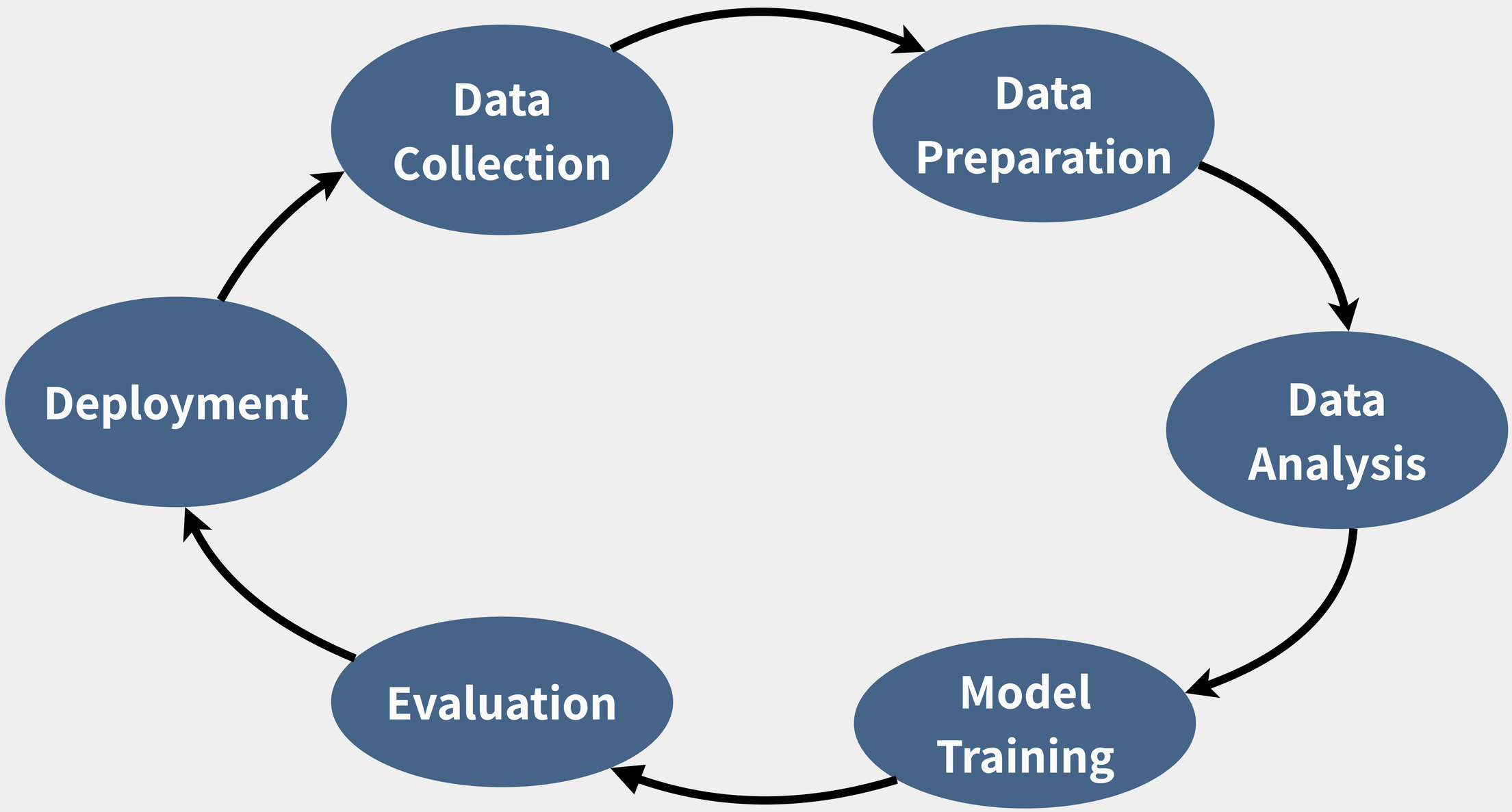

baths
price
1e7
sqft
beds
baths

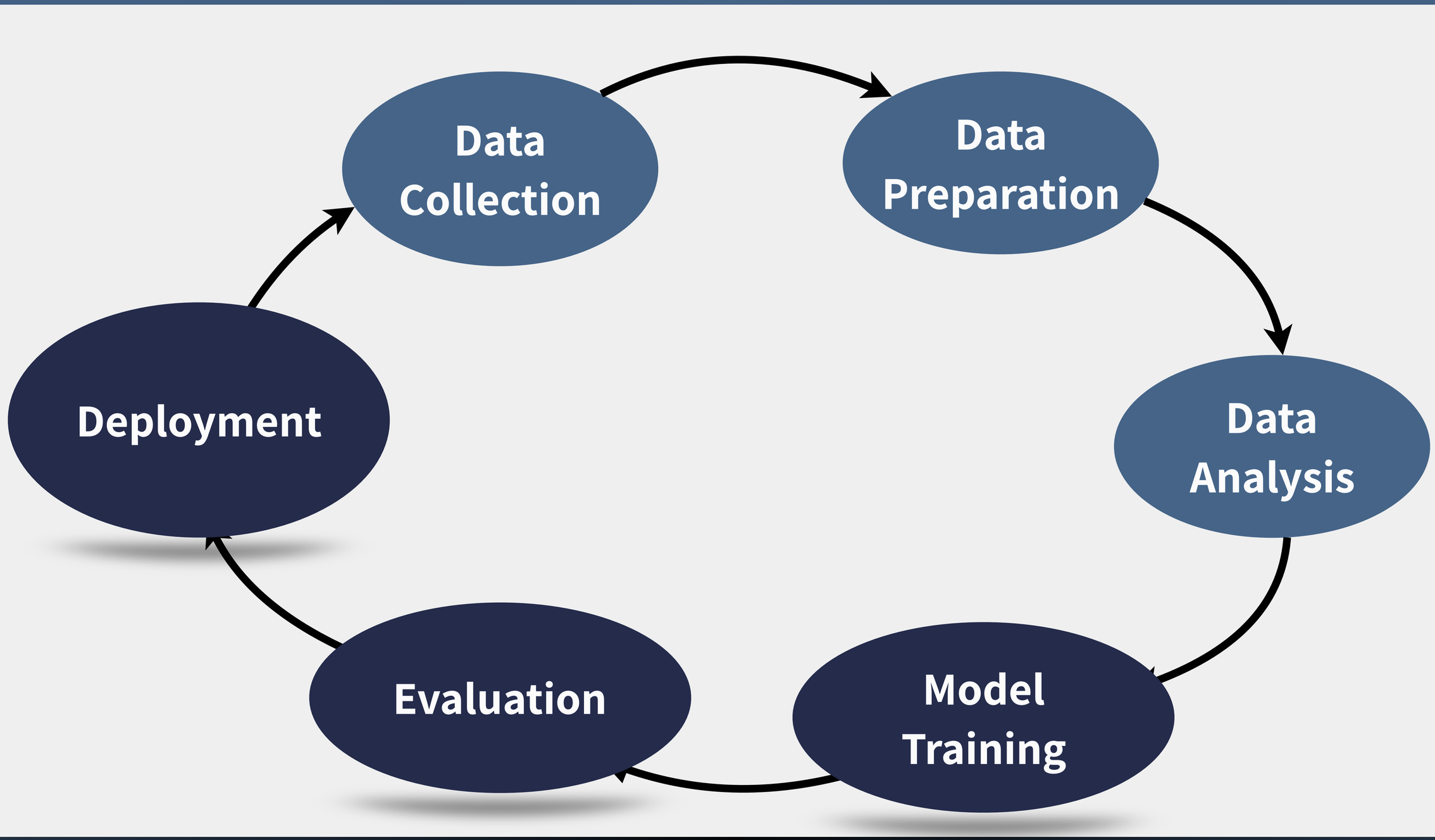
Deployment

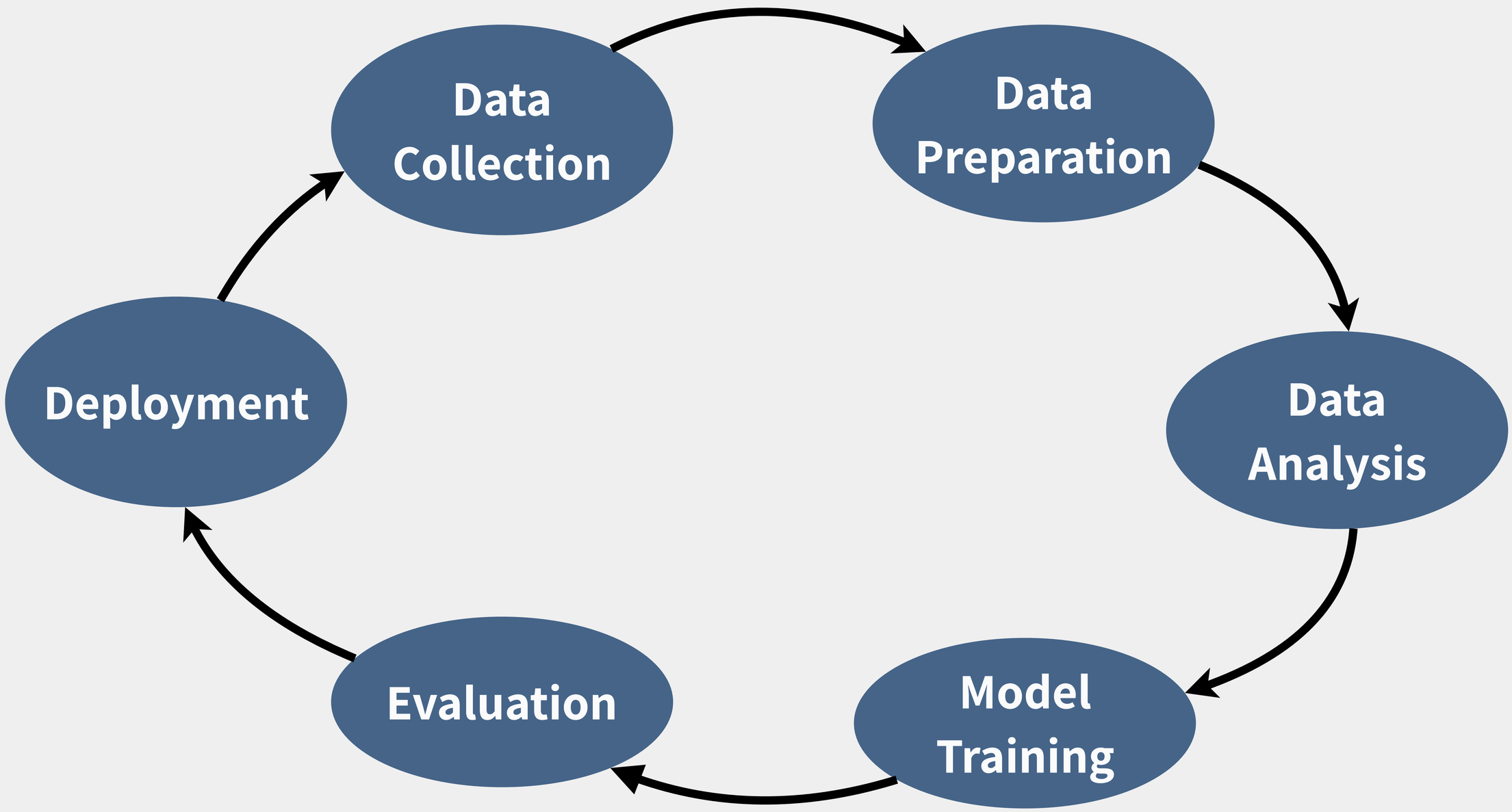
Data Analysis




2500
2000
1500
1000
500
0
count
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
beds


Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment


Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment


Data Collection
Data Preparation
Data Analysis
Model Training
Evaluation
Deployment

Data Collection

Data Preparation

Deployment

Data Analysis

Evaluation

Model Training

Types of ML Algorithms

Types of ML Algorithms

Supervised Learning
Unsupervised Learning
Reinforcement Learning

Types of ML Algorithms

Supervised Learning

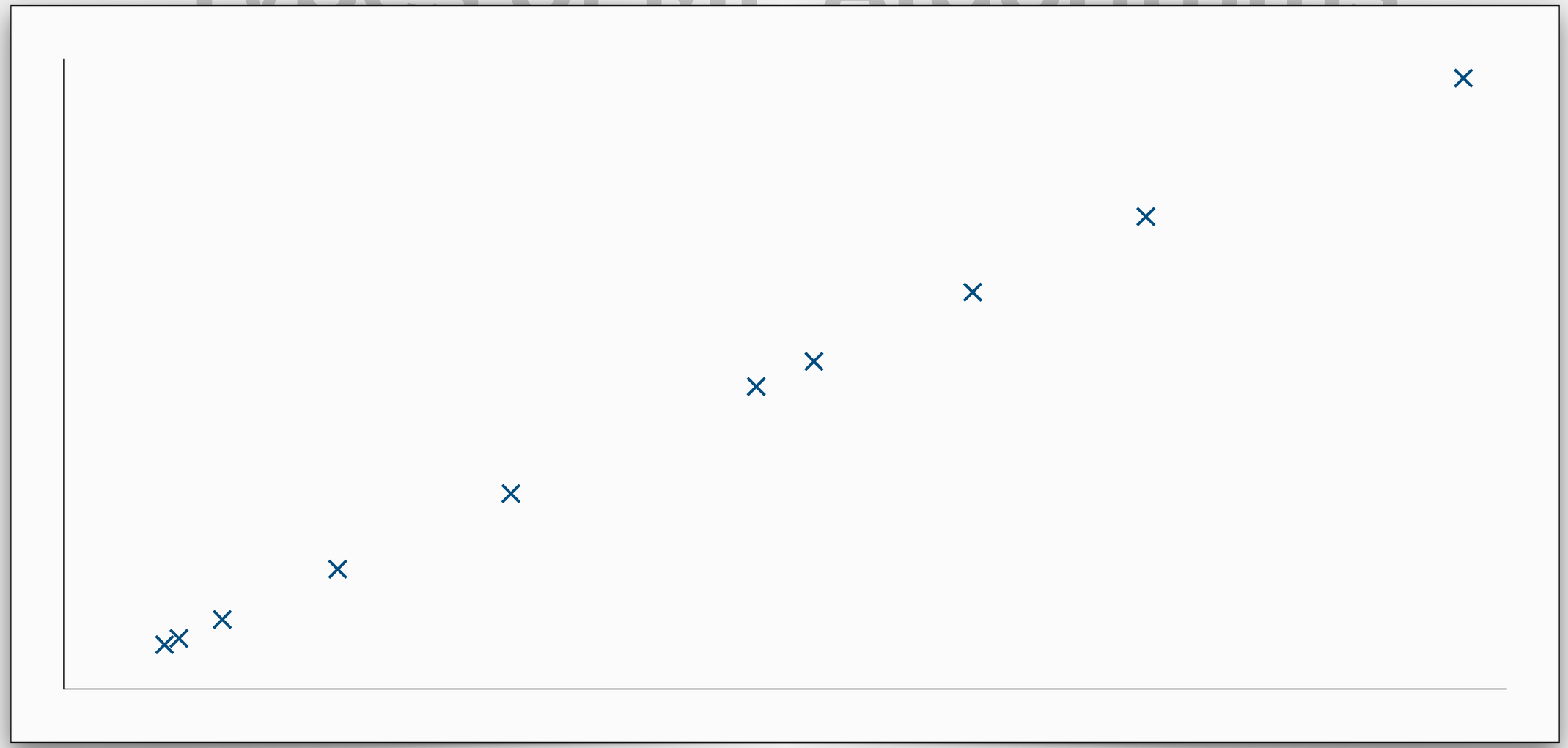
Classification
Regression

Types of ML Algorithms

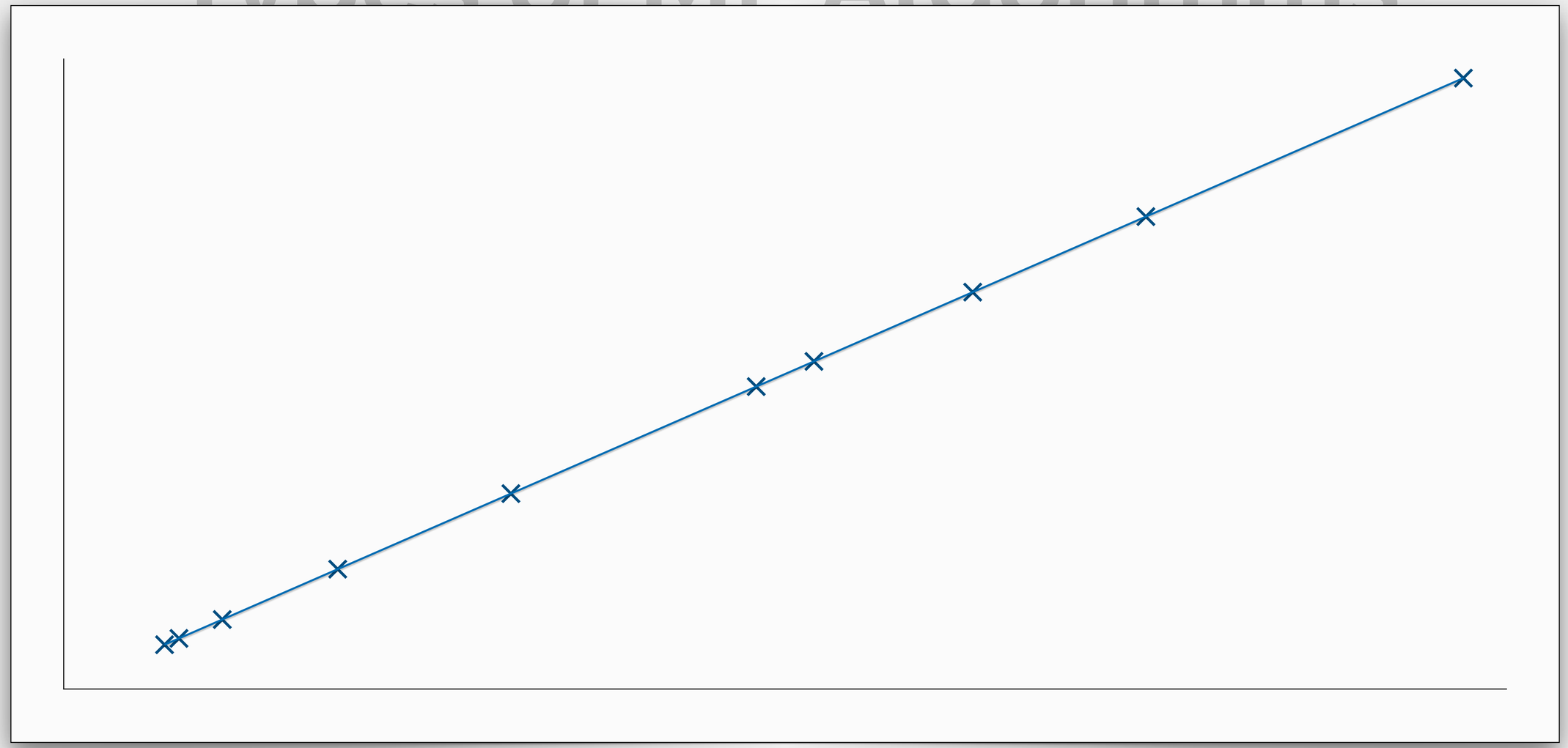
Supervised Learning

Regression

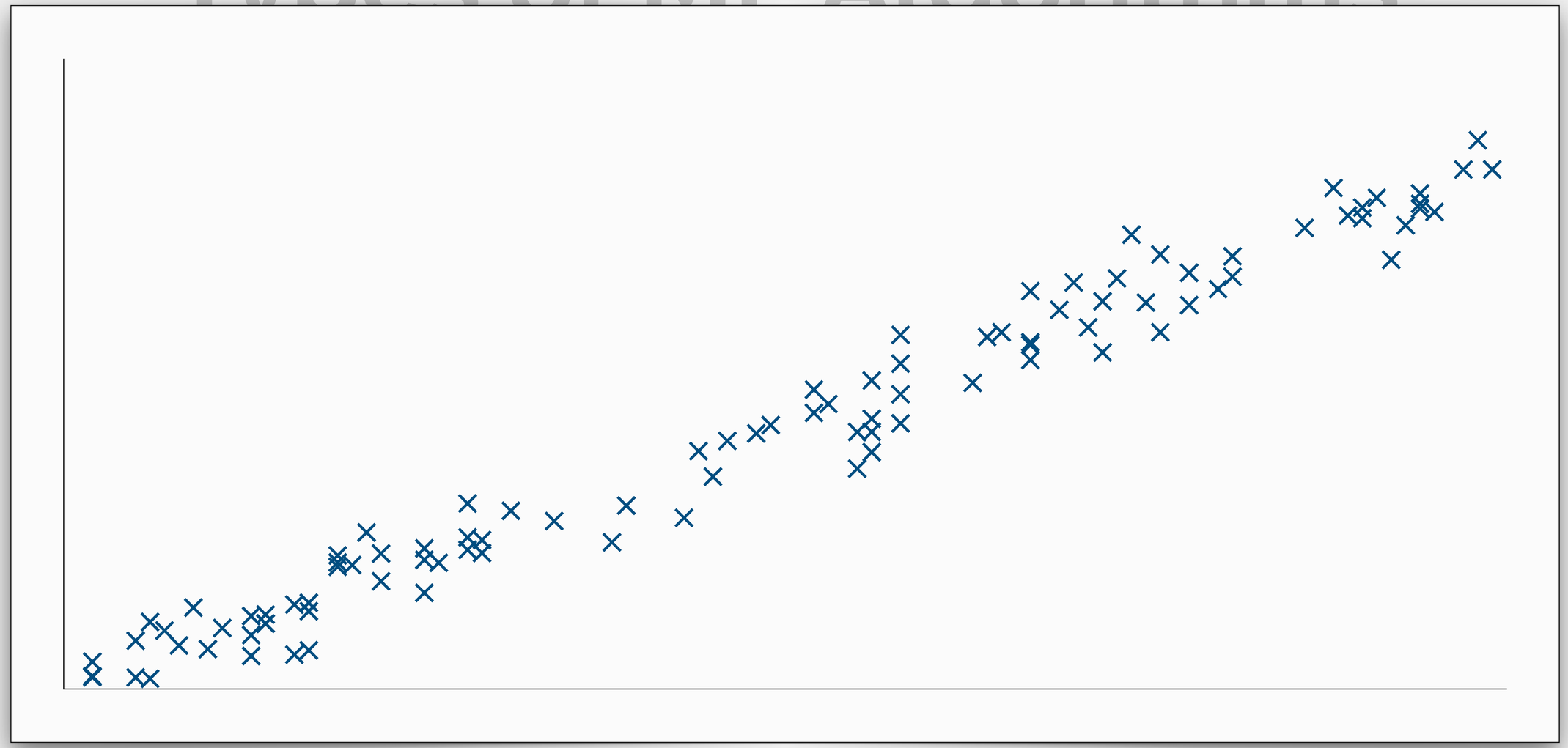
Types of ML Algorithms



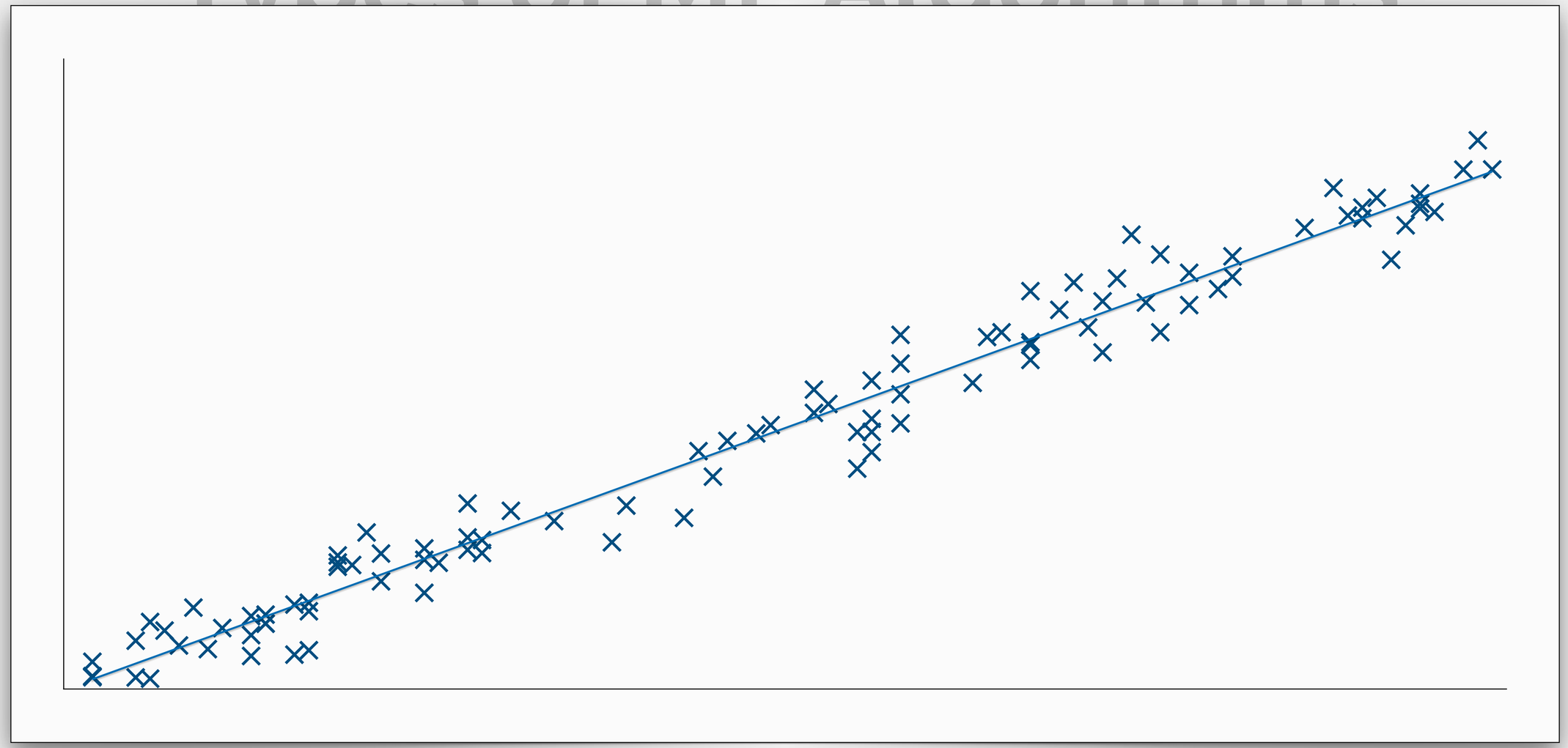
Types of ML Algorithms



Types of ML Algorithms



Types of ML Algorithms



Types of ML Algorithms

Supervised Learning
Unsupervised Learning
Reinforcement Learning

Types of ML Algorithms

Unsupervised Learning

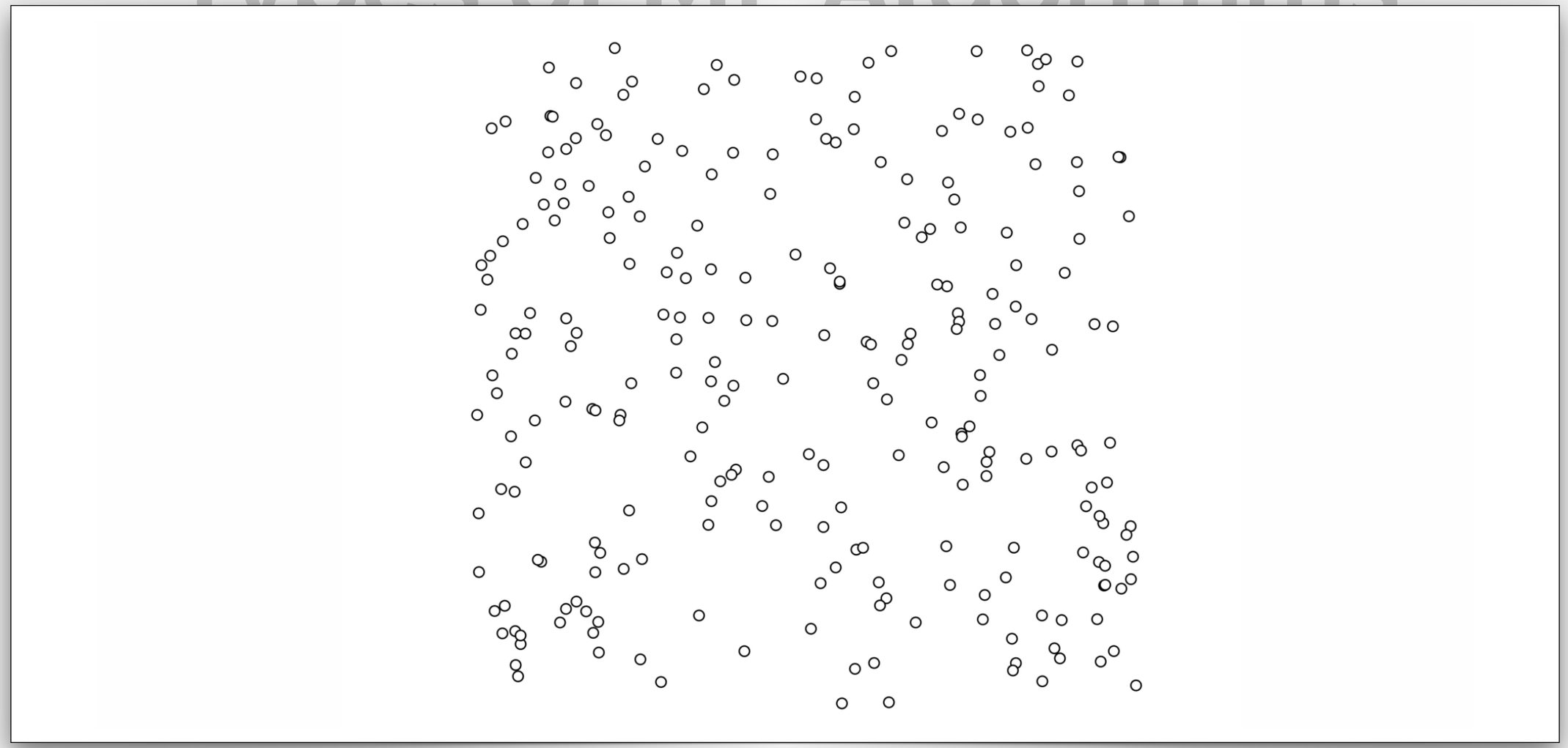
Clustering
Association

Types of ML Algorithms

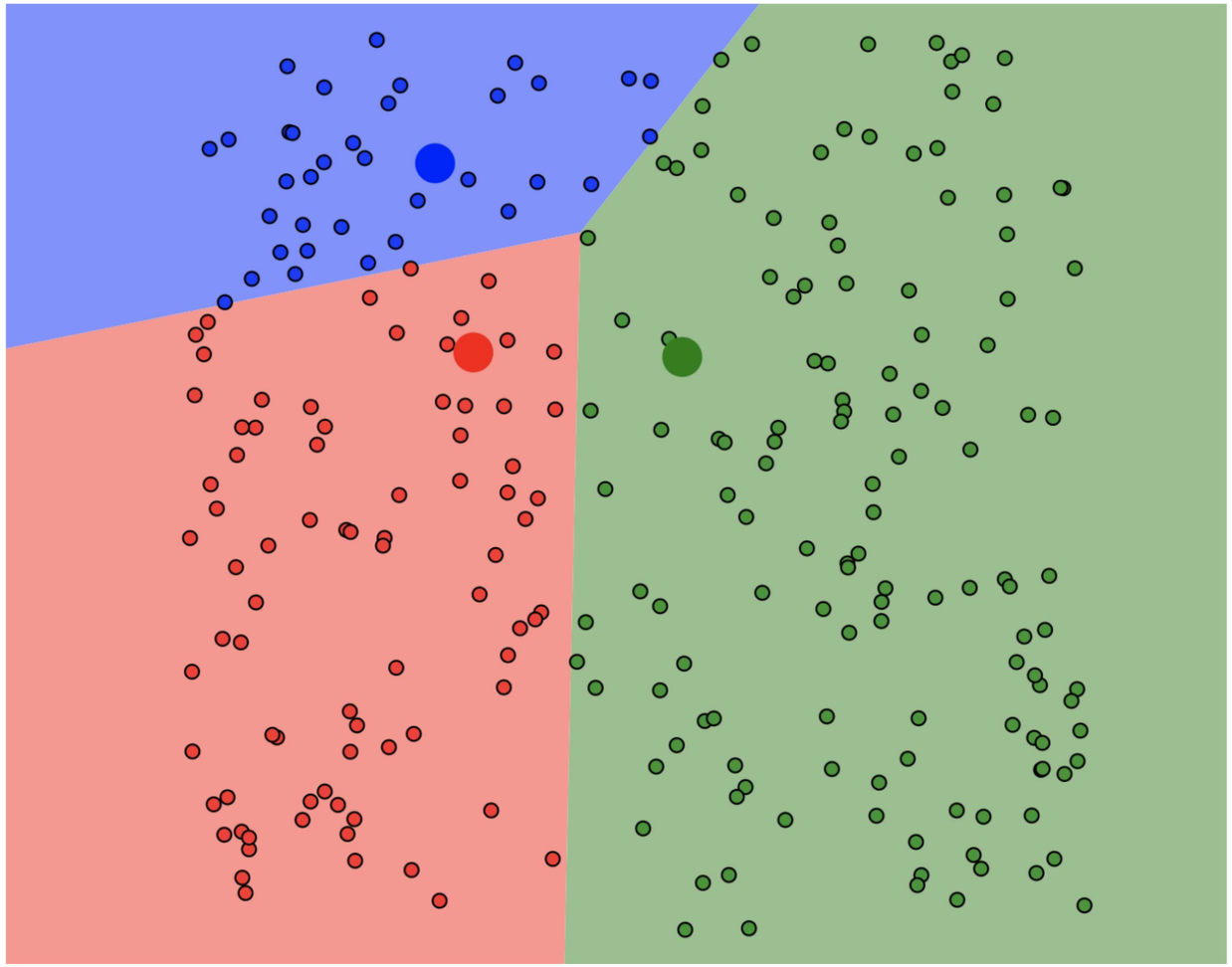
Unsupervised Learning

Clustering

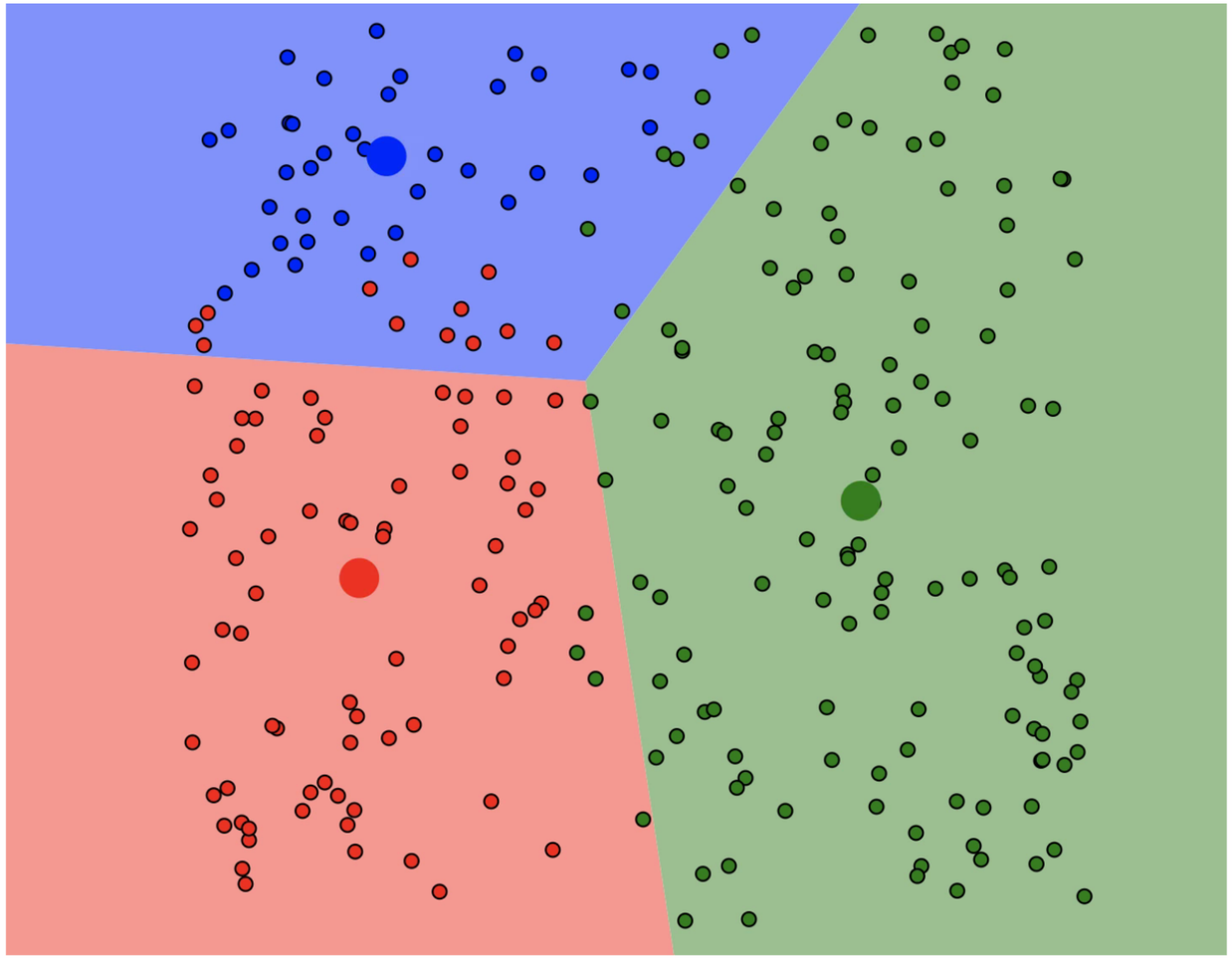
Types of ML Algorithms



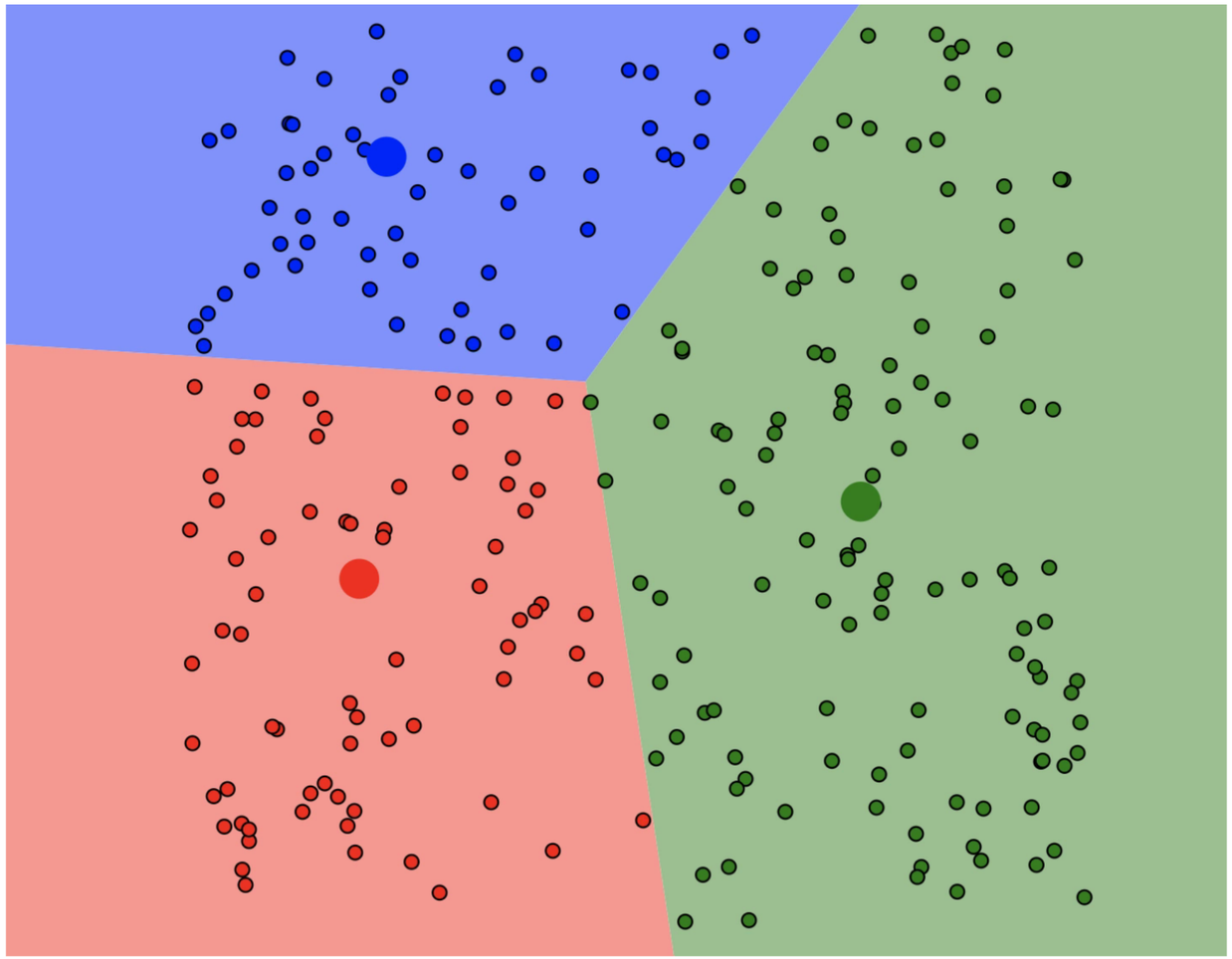
Types of ML Algorithms



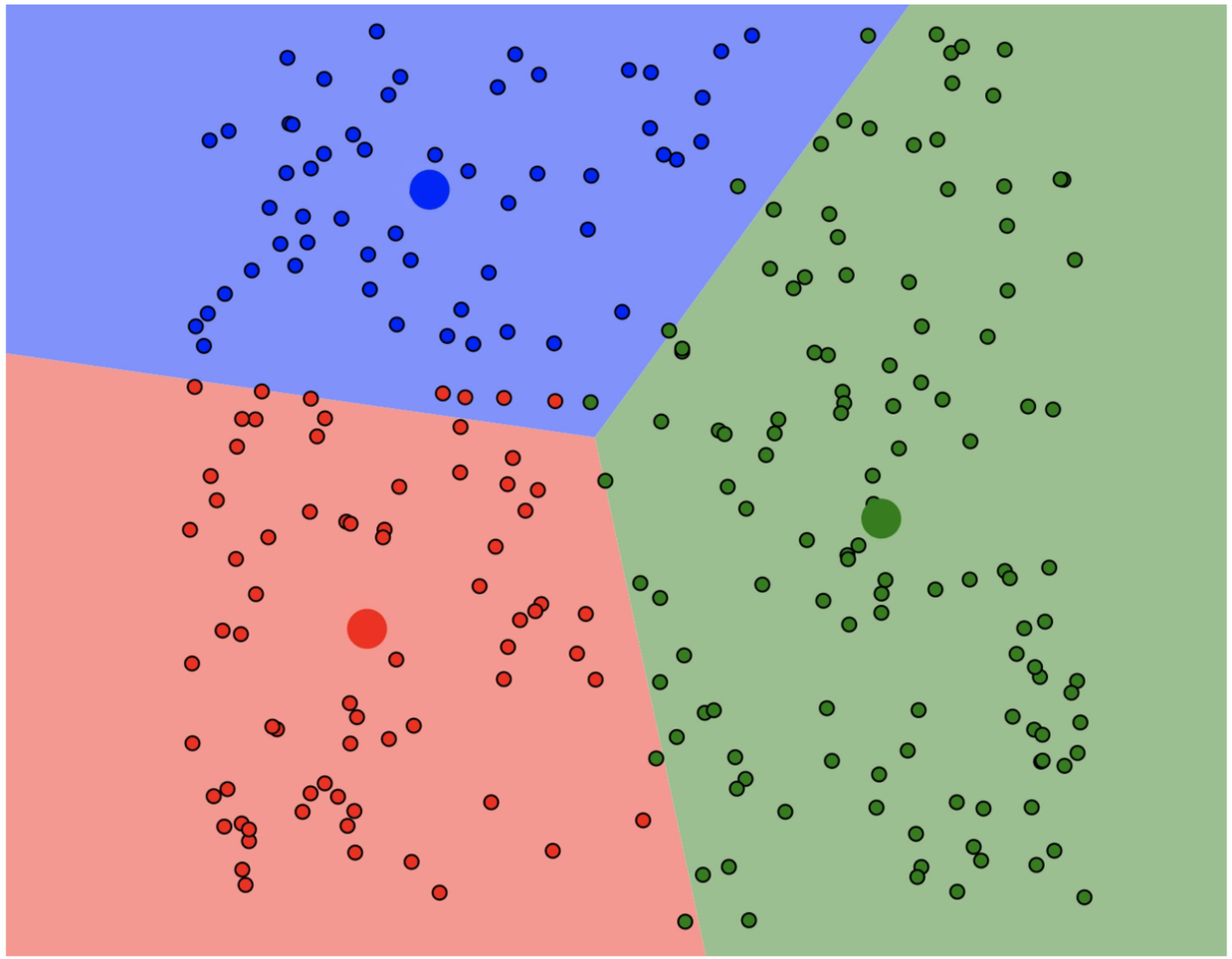
Types of ML Algorithms



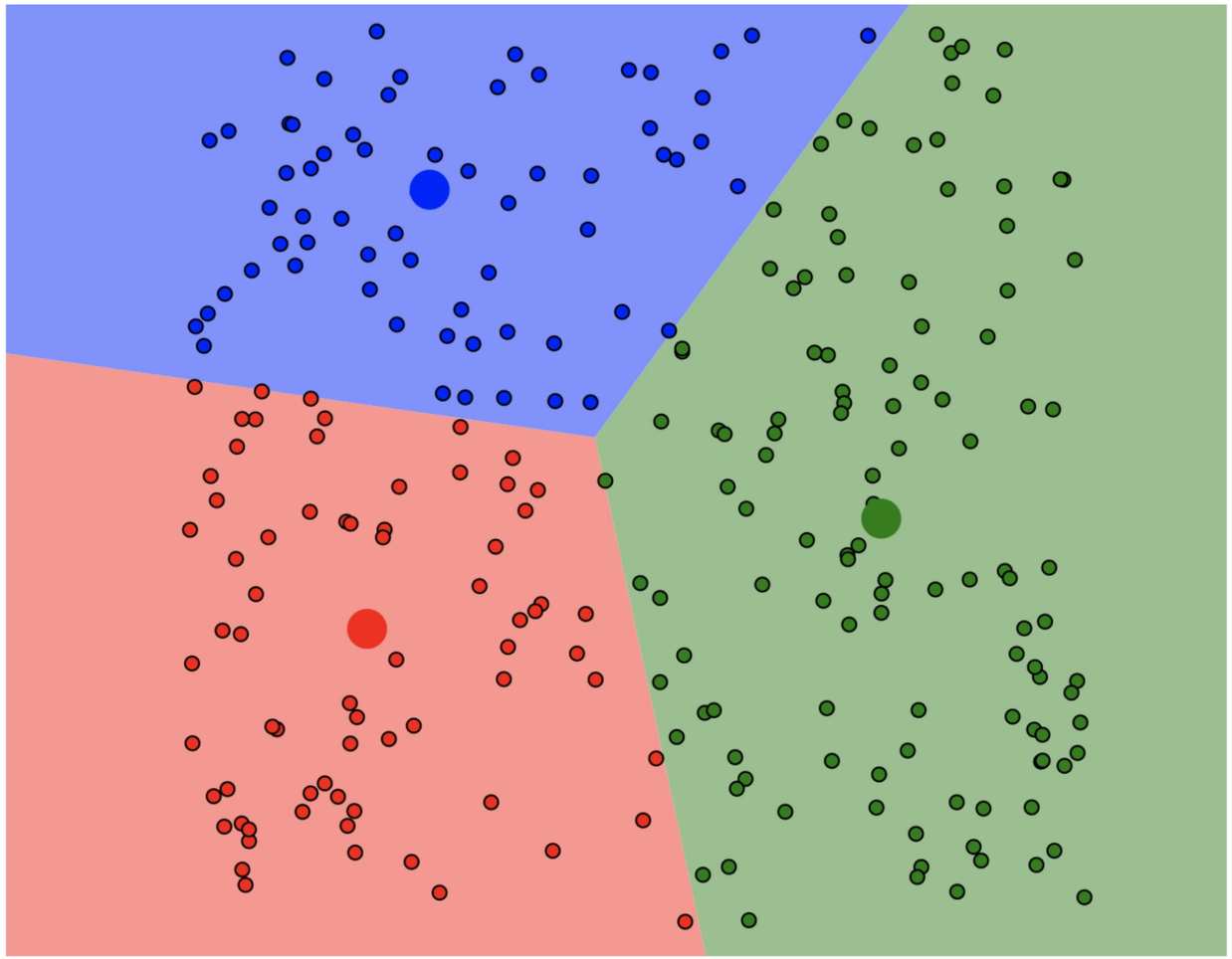
Types of ML Algorithms



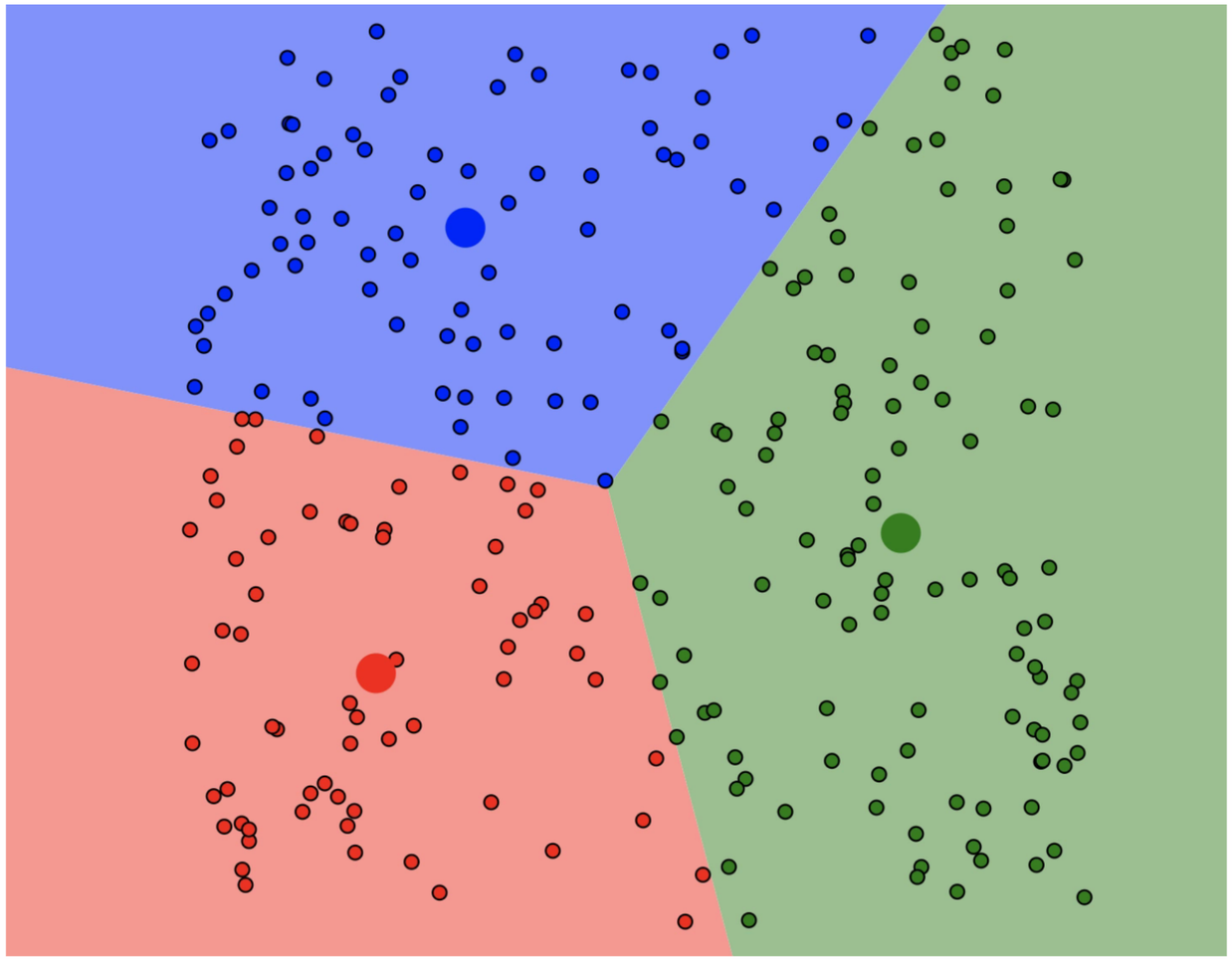
Types of ML Algorithms



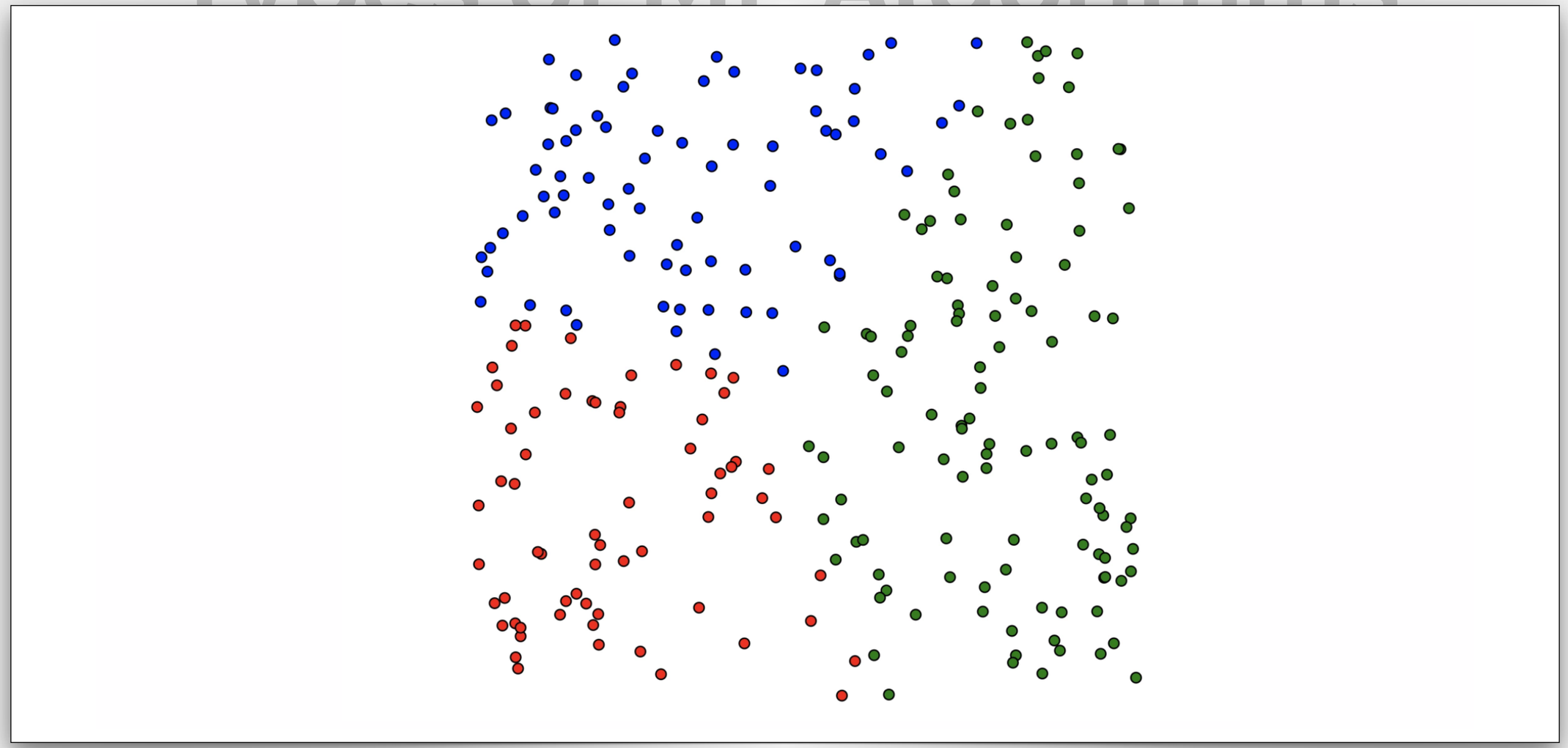
Types of ML Algorithms



Types of ML Algorithms



Types of ML Algorithms



Types of ML Algorithms

Supervised Learning
Unsupervised Learning
Reinforcement Learning

Types of ML Algorithms

Reinforcement Learning

Reward Driven

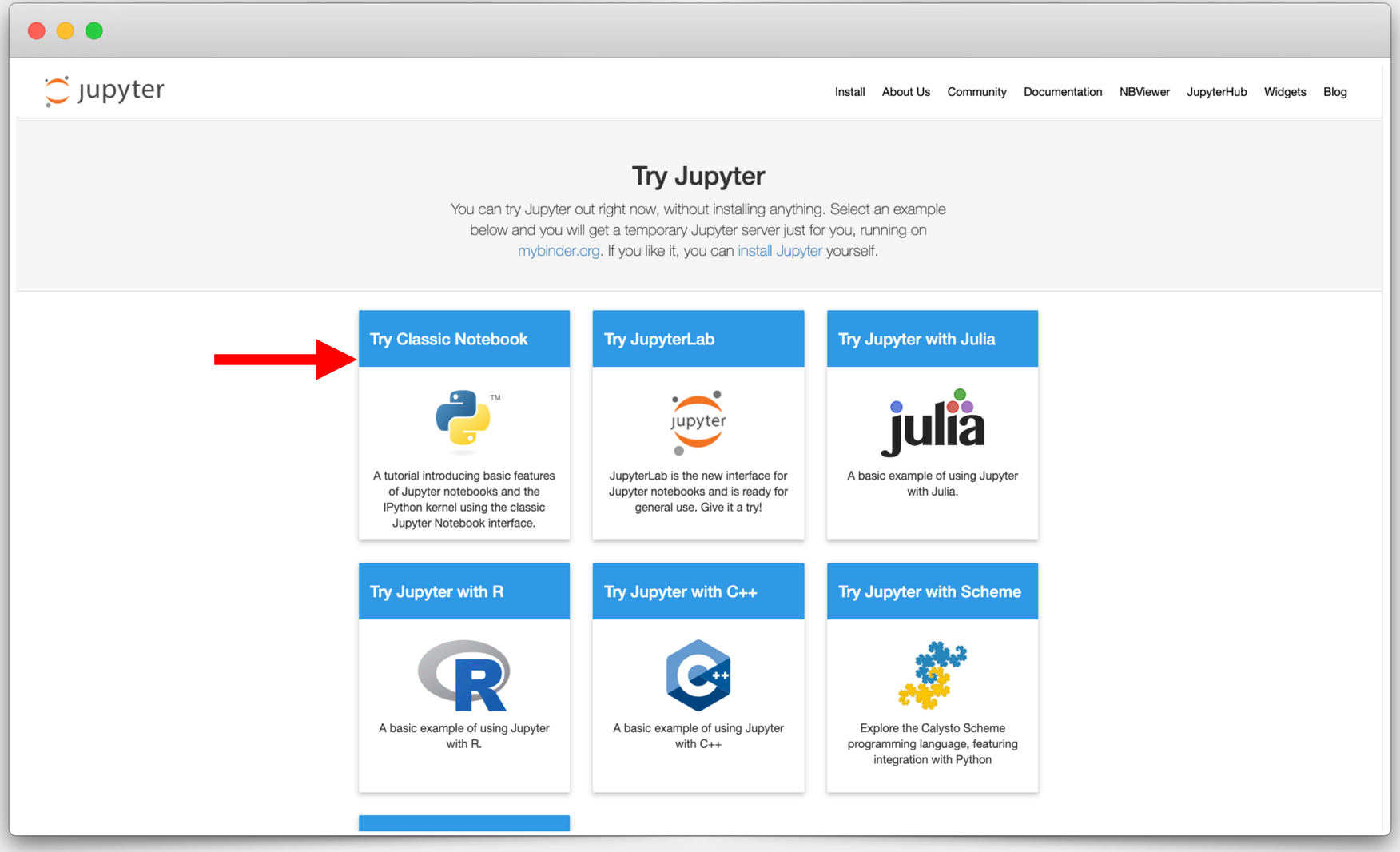
Demo


jupyter

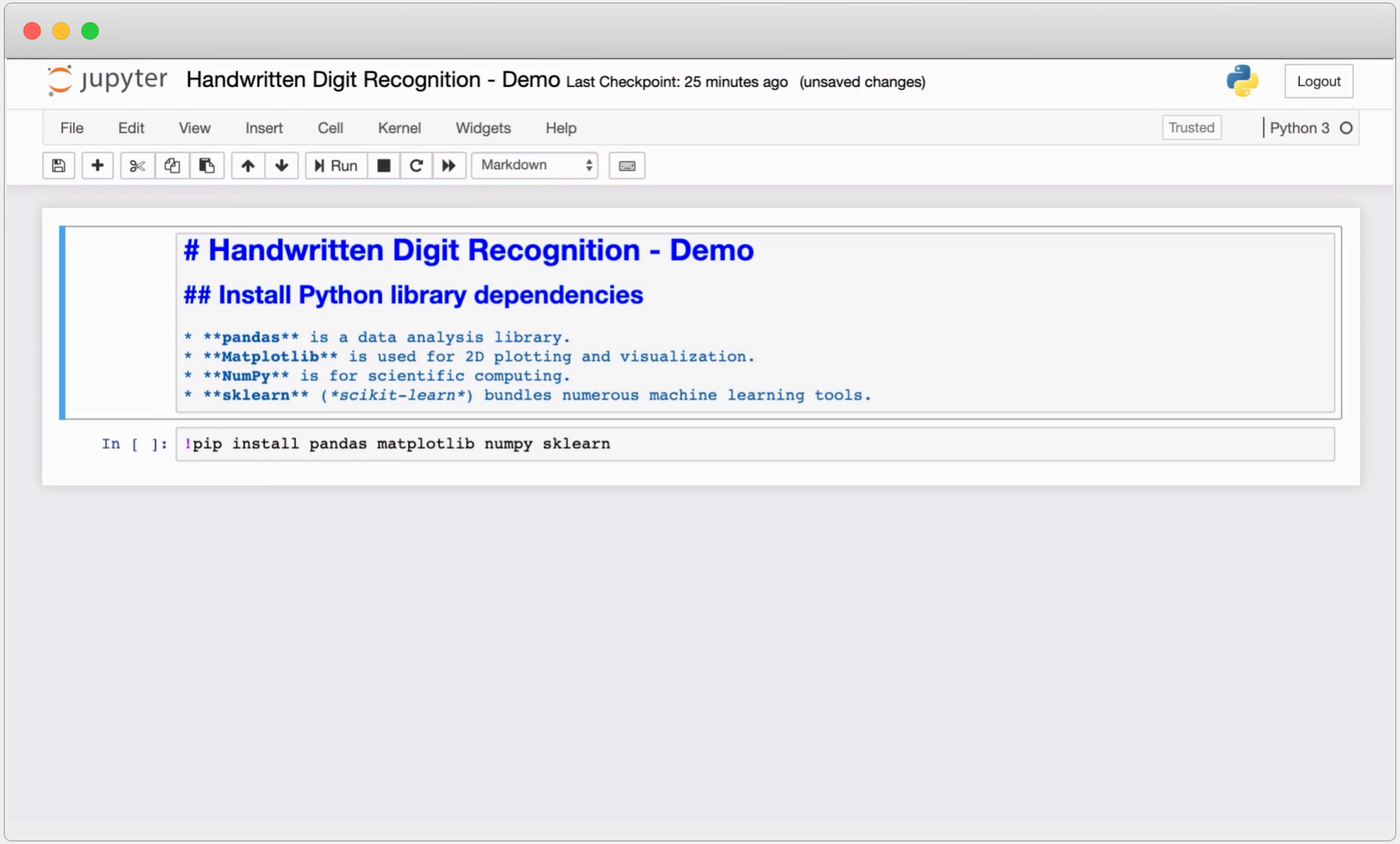
https://jupyter.readthedocs.io/en/latest/install.html

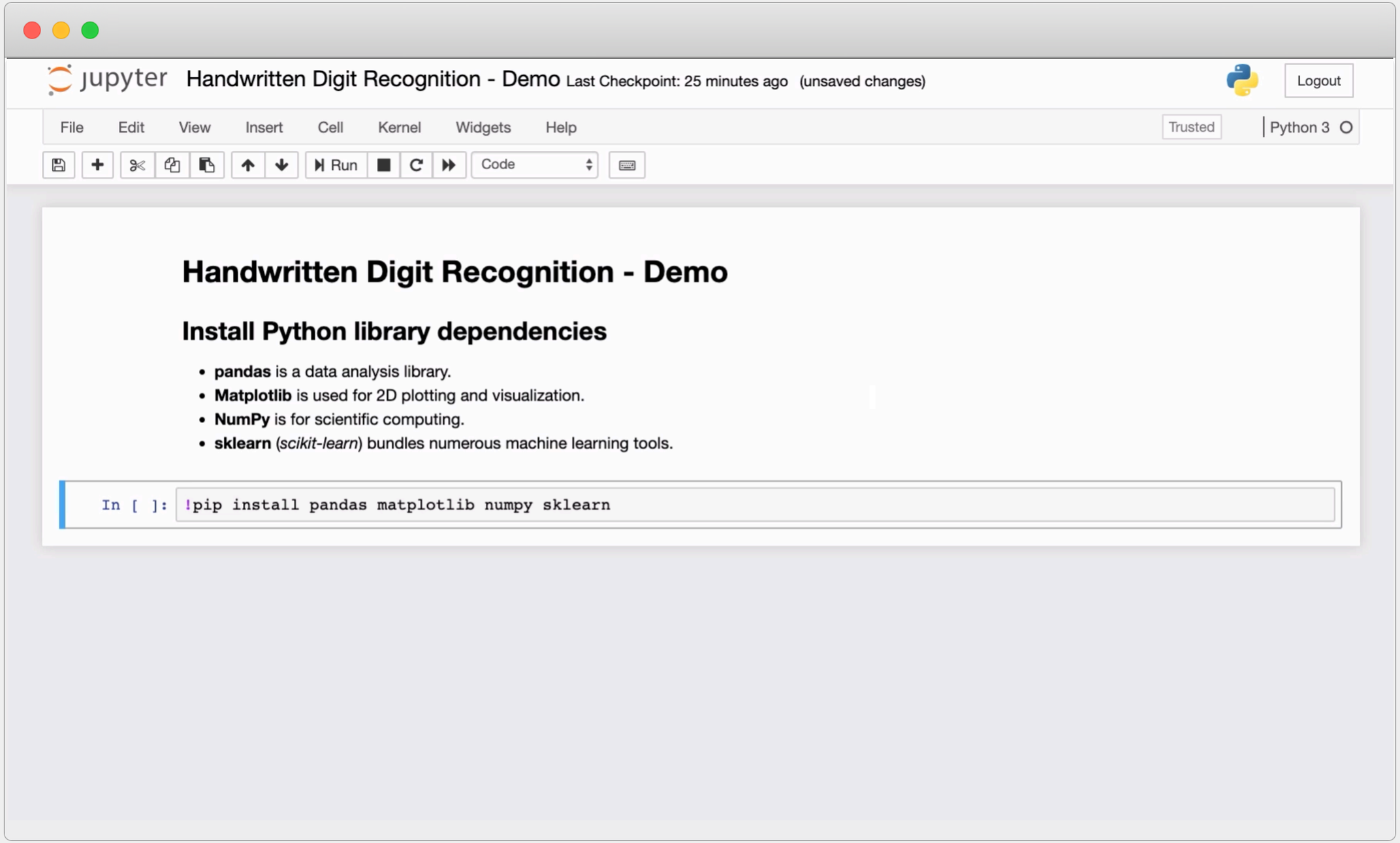
https://jupyter.readthedocs.io/en/latest/install.html

https://jupyter.org/try



https://jupyter.org/try


jupyter Handwritten Digit Recognition - Demo Last Checkpoint: 25 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Markdown
Handwritten Digit Recognition - Demo
Install Python library dependencies
* **pandas** is a data analysis library.
* **Matplotlib** is used for 2D plotting and visualization.
* **NumPy** is for scientific computing.
* **sklearn** (*scikit-learn*) bundles numerous machine learning tools.
In [ ]:
!pip install pandas matplotlib numpy sklearn


jupyter Handwritten Digit Recognition - Demo Last Checkpoint: 25 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Handwritten Digit Recognition - Demo
Install Python library dependencies
pandas is a data analysis library.
Matplotlib is used for 2D plotting and visualization.
NumPy is for scientific computing.
sklearn (scikit-learn) bundles numerous machine learning tools.
In [ ]:
!pip install pandas matplotlib numpy sklearn

```
!wget http://deeplearning.net/data/mnist/mnist.pkl.gz -O mnist.pkl.gz
!gunzip mnist.pkl.gz -vf


--2019-09-16 13:01:24-- http://deeplearning.net/data/mnist/mnist.pkl.gz
Resolving deeplearning.net (deeplearning.net)... 132.204.26.28
Connecting to deeplearning.net (deeplearning.net)|132.204.26.28|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 16168813 (15M) [application/x-gzip]
Saving to: ‘mnist.pkl.gz’

mnist.pkl.gz 100%[===================>] 15.42M 4.60MB/s in 3.5s

2019-09-16 13:01:28 (4.39 MB/s) - ‘mnist.pkl.gz’ saved [16168813/16168813]

mnist.pkl.gz: 92.6% -- replaced with mnist.pkl
```

```
import pandas as pd

(train_set,
validation_set,
test_set) = pd.read_pickle('./mnist.pkl')
```

```
print(train_set)
```

```
(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.]],
dtype=float32), array([5, 0, 4, ..., 8, 4, 8]))
```

```
len(train_set[0]), len(train_set[1])
```

```
(50000, 50000)
```

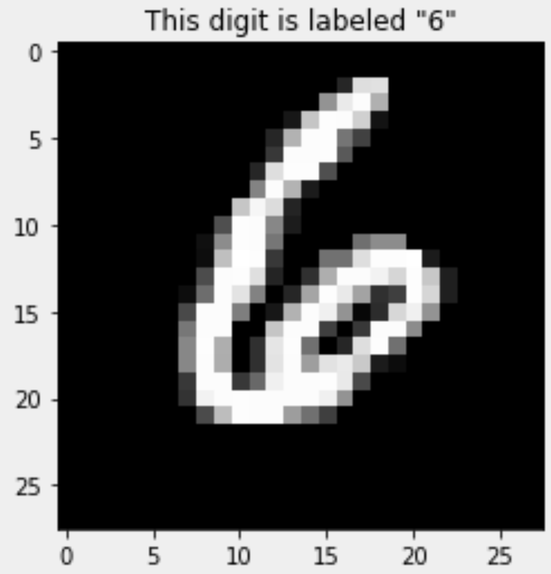
```
print(train_set[1][13])
```

```
6
```

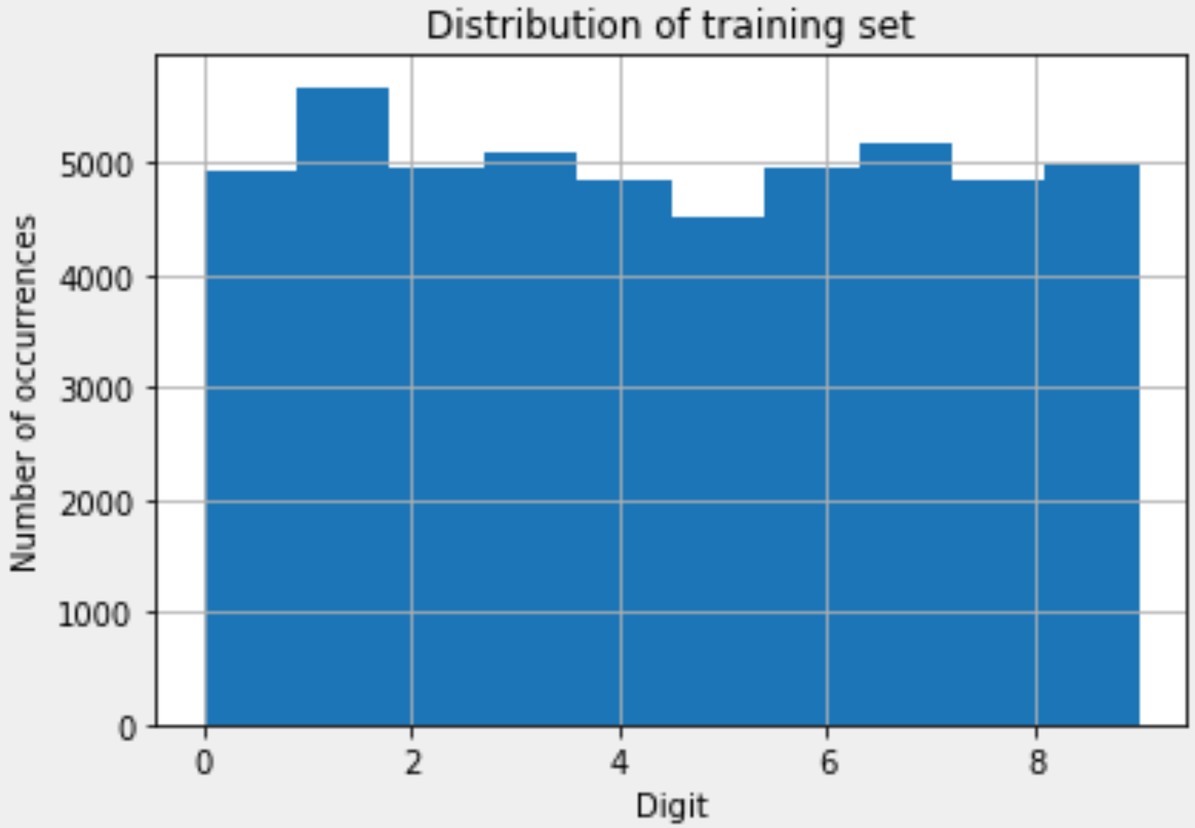
```
print(train_set[0][13])
```

```
0.45703125 0.25390625 0.         0.         0.         0.
0.         0.         0.         0.         0.         0.
0.         0.         0.         0.         0.         0.
0.         0.         0.         0.         0.         0.
0.22265625 0.984375   0.984375   0.98828125 0.34765625 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.         0.1484375  0.8671875  0.98828125
0.98828125 0.30859375 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.         0.51171875 0.984375   0.69921875 0.10546875 0.
0.         0.         0.         0.         0.         0.
```

```
import matplotlib.pyplot as plt

image_data = train_set[0][13].reshape((28, 28))
image_label = train_set[1][13]

plt.title('This digit is labeled "{}"'
     .format(image_label))

plt.imshow(image_data, cmap='gray')
```



```
plt.hist(train_set[1])
plt.title('Distribution of training set')
plt.ylabel('Number of occurrences')
plt.xlabel('Digit')
plt.grid(True)
```



```
%%time
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import accuracy_score

clf_sgd = SGDClassifier()
clf_sgd.fit(train_set[0], train_set[1])
clf_sgd_predictions = clf_sgd.predict(validation_set[0])

print('Training Accuracy: ',
      clf_sgd.score(train_set[0], train_set[1]))
print('Validation Accuracy: ',
    accuracy_score(validation_set[1], clf_sgd_predictions))
```

```
%%time
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import accuracy_score

print('Validation Accuracy: ',
    accuracy_score(validation_set[1], clf_sgd_predictions))
```

```
Training Accuracy:  0.91918
Validation Accuracy:  0.9164
CPU times: user 13.7 s, sys: 519 ms, total: 14.3 s
Wall time: 14.3 s
```

```
%%time
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

clf_rf = RandomForestClassifier()
clf_rf.fit(train_set[0], train_set[1])
clf_rf_predictions = clf_rf.predict(validation_set[0])

print('Training Accuracy: ',
    clf_rf.score(train_set[0], train_set[1]))
print('Validation Accuracy: ',
    accuracy_score(validation_set[1], clf_rf_predictions))
```

```
%%time
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

print('Validation Accuracy: ',
    accuracy_score(validation_set[1], clf_rf_predictions))
```

```
Training Accuracy:  0.9994
Validation Accuracy:  0.9513
CPU times: user 2.92 s, sys: 45.6 ms, total: 2.97 s
Wall time: 3.16 s
```

```
print('Test Accuracy (SGD): ',
    accuracy_score(
        test_set[1], clf_sgd.predict(test_set[0])
    )
)

print('Test Accuracy (RandomForest): ',
    accuracy_score(
        test_set[1], clf_rf.predict(test_set[0])
    )
)
```

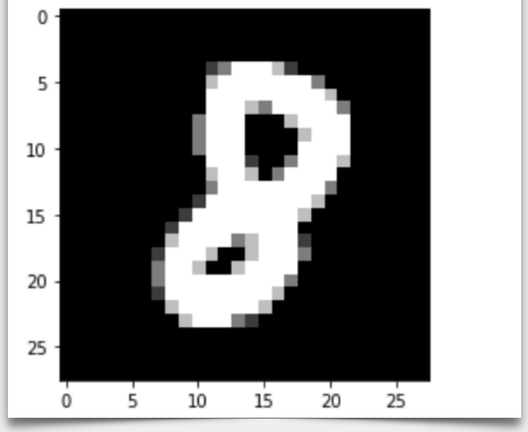
```
Test Accuracy (SGD):  0.9163
Test Accuracy (RandomForest):  0.9437
```

```
import numpy as np

indices = np.random.randint(
    low=1, high=len(test_set[0]), size=5)

for index in indices:
    image = test_set[0][index]
    image_label = test_set[1][index]

    plt.imshow(image.reshape((28, 28)), cmap='gray')

    print('Predicting digit at index: {}'
        .format(index))
```

```
print('Predicting digit at index: {}'
      .format(index))

plt.title(
      'SGDClassifier guessed: {}\n'
          .format(clf_sgd.predict([image])[0]) +
      'RandomForestClassifier guessed: {}\n'
          .format(clf_rf.predict([image])[0]) +
      'The correct answer is: {}\n'
          .format(image_label),

      loc='left'
)

plt.show()
```

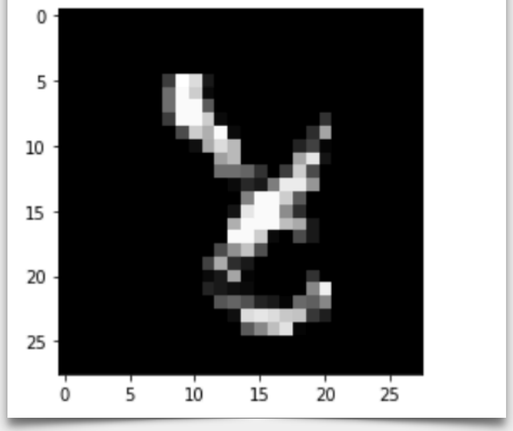
```
Predicting digit at index 7287
```

SGDClassifier guessed this is a: 8
RandomForestClassifier guessed this is a: 8
The correct answer is: 8



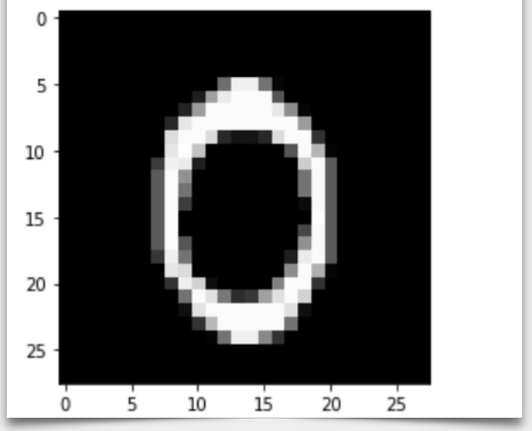
```
Predicting digit at index 6625
```

SGDClassifier guessed this is a: 4
RandomForestClassifier guessed this is a: 7
The correct answer is: 8



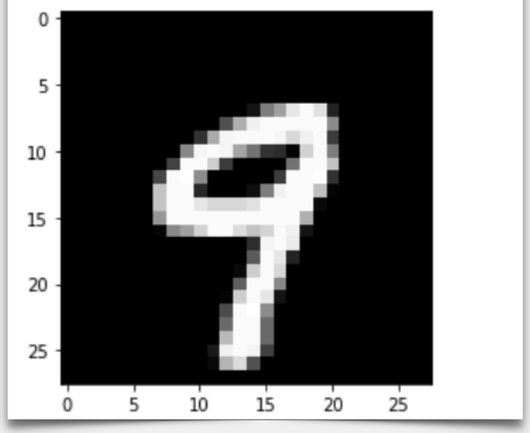
```
Predicting digit at index 3322
```

SGDClassifier guessed this is a: 0
RandomForestClassifier guessed this is a: 0
The correct answer is: 0



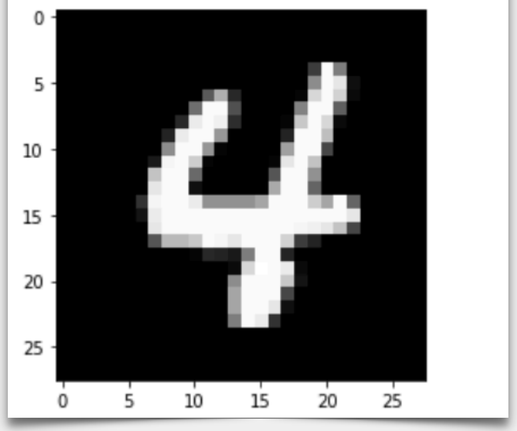
```
Predicting digit at index 8705
```

SGDClassifier guessed this is a: 9
RandomForestClassifier guessed this is a: 9
The correct answer is: 9

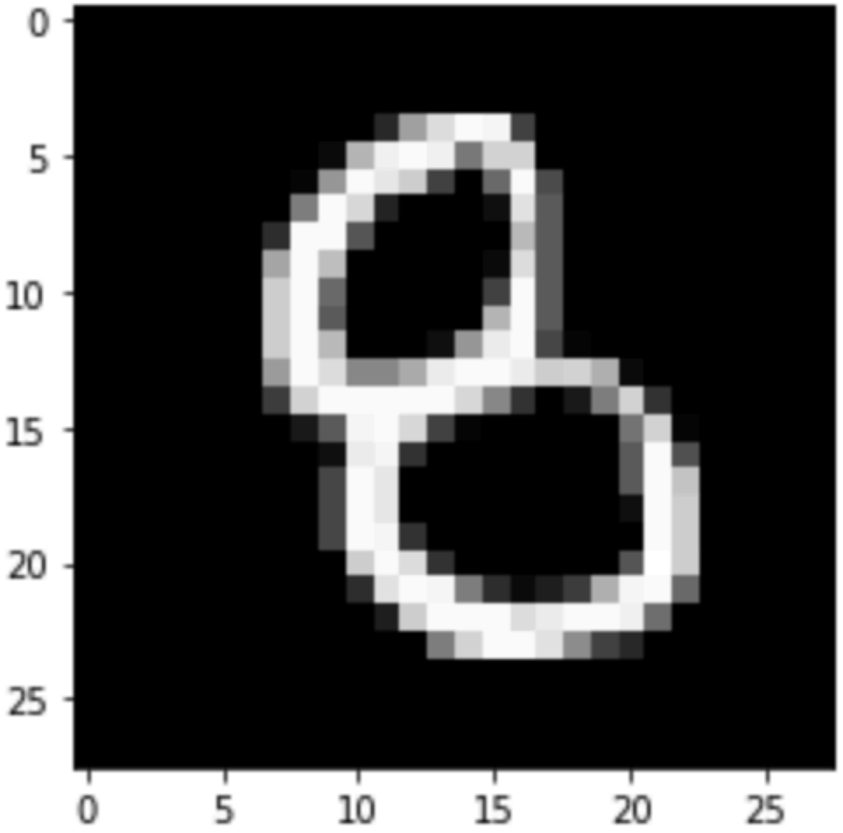


```
Predicting digit at index 8790
```

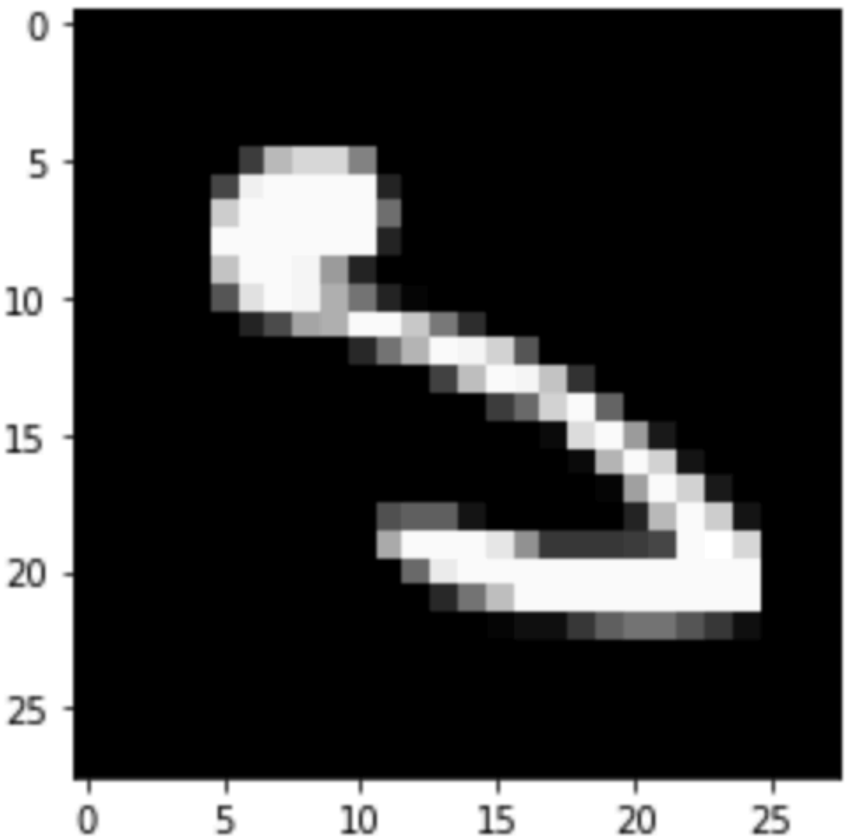
SGDClassifier guessed this is a: 4
RandomForestClassifier guessed this is a: 4
The correct answer is: 4



SGDClassifier guessed this is a: 5
RandomForestClassifier guessed this is a: 8
The correct answer is: 8



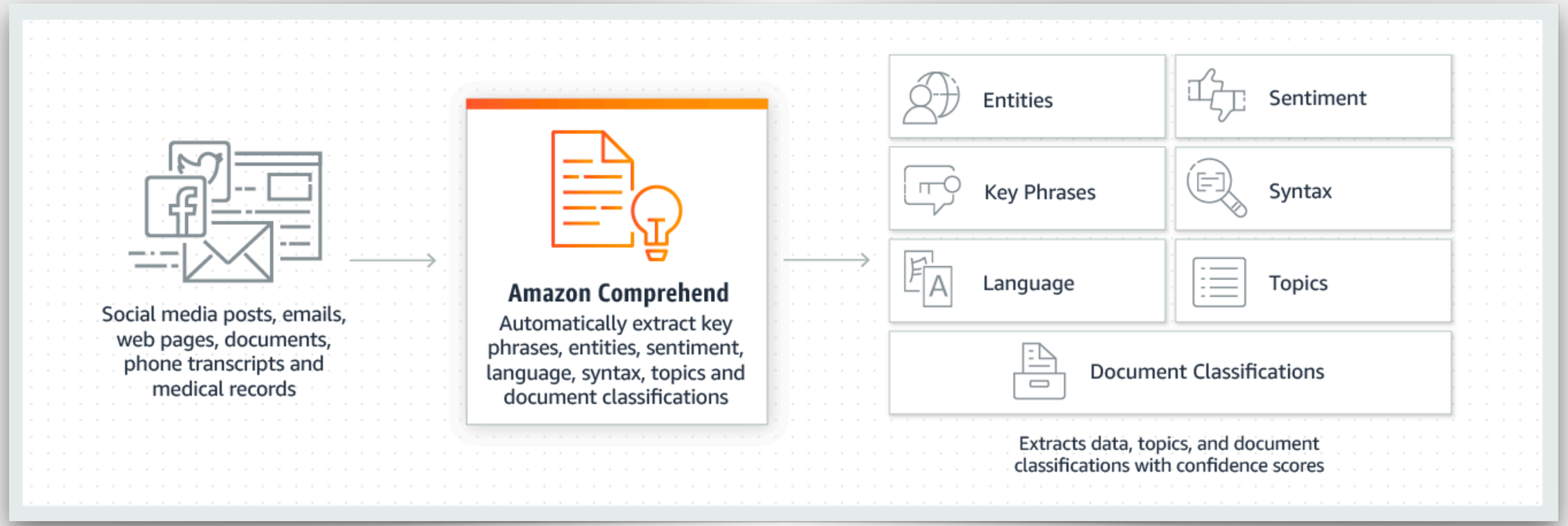
SGDClassifier guessed this is a: 5
RandomForestClassifier guessed this is a: 3
The correct answer is: 5



Machine Learning on AWS

Machine Learning on AWS

Amazon Comprehend



Machine Learning on AWS

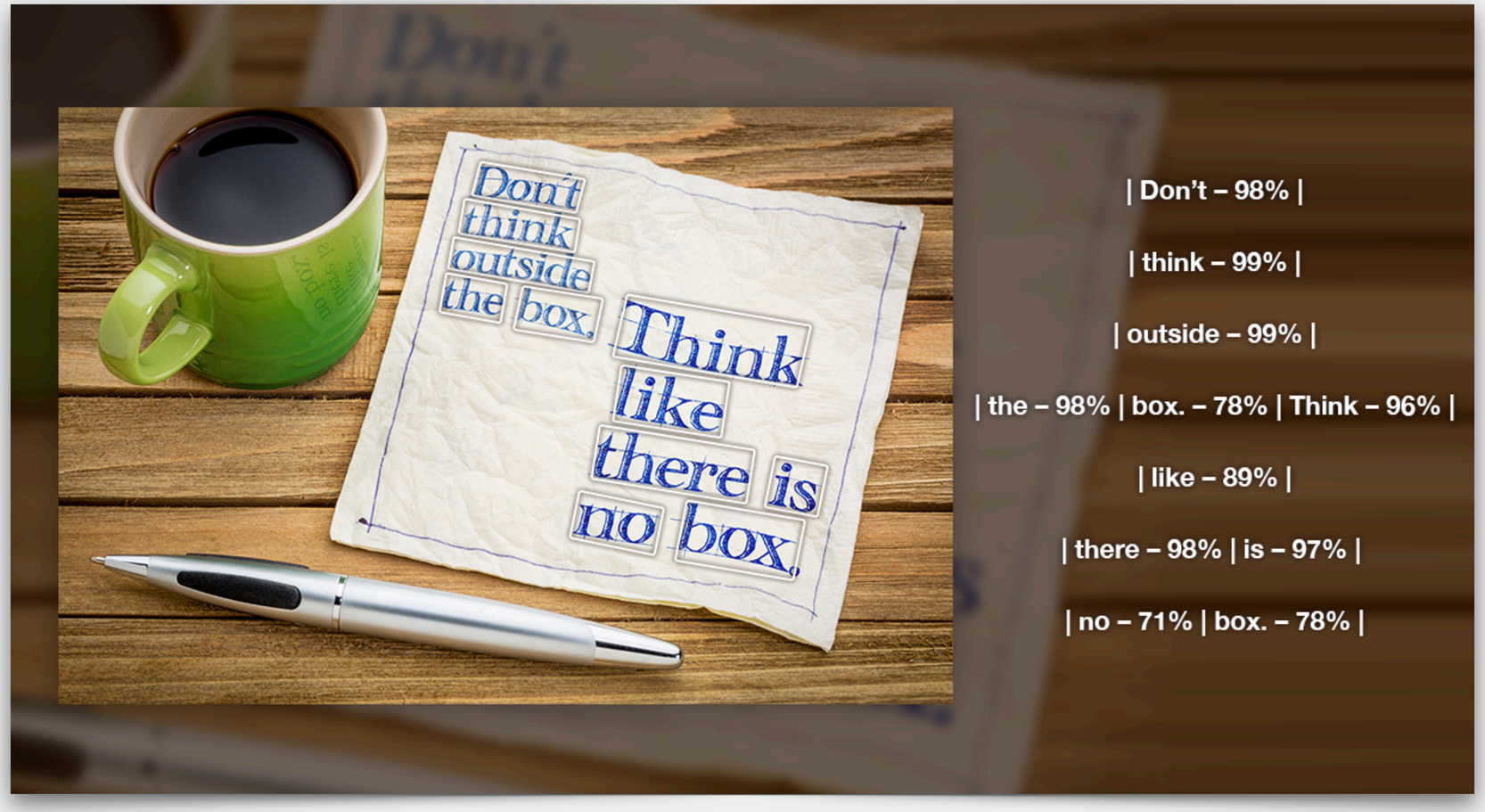
Amazon Personalize

Machine Learning on AWS

Amazon Rekognition

Machine Learning on AWS

Amazon Rekognition



Machine Learning on AWS

Amazon Rekognition

Machine Learning on AWS

https://aws.amazon.com/machine-learning/

Thanks!

https://joind.in/talk/5f55e