What should I know about Cross-Validation as a Data Science Beginner?

Abraham Kong
CMPE 255-49

Why should we know about Cross-Validation?

- Many Machine Learning Method to choose from as a Data Scientist

- Need to know which method performs better
  Since each method has its own benefits and its own biased

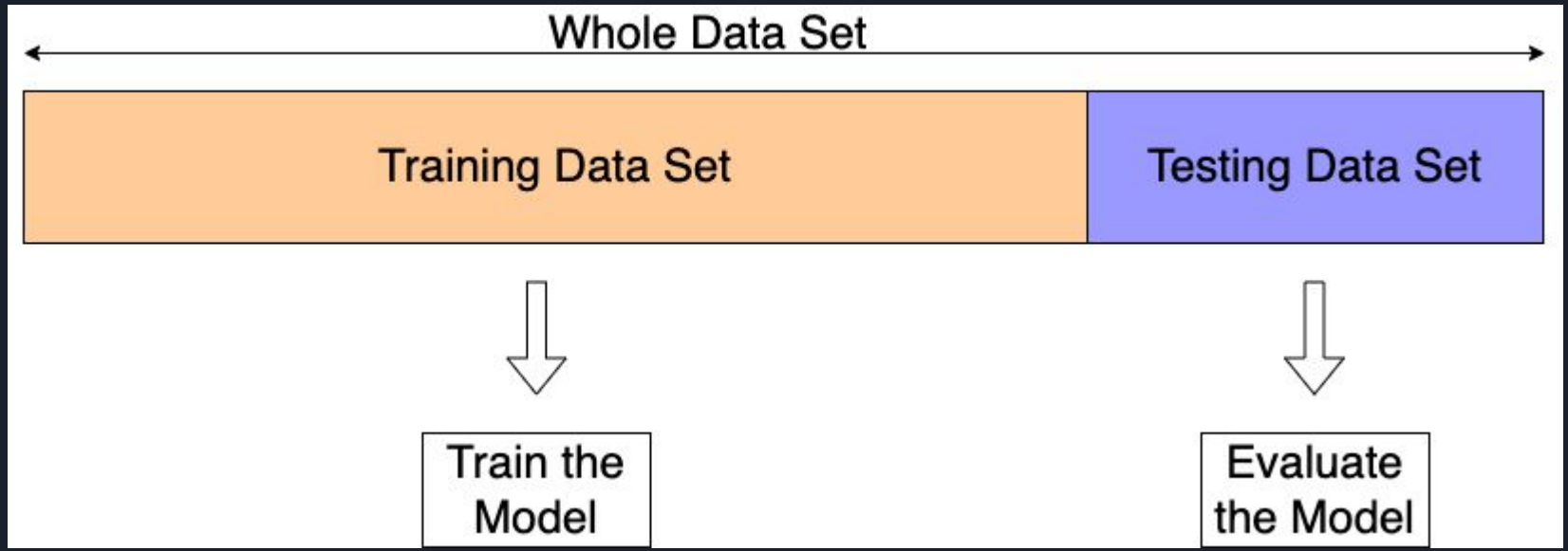
- Cross- Validation allows Data Scientist to evaluate different Machine Learning Method and choose the one fit the data the best

What is Cross-Validation?

- Cross Validation is a “re-sampling” technique that based on the idea of splitting data into “Training Data Set and “Testing Data Set”

- We use the “Training Data Set” to train the model, and use the “Testing Data Set” to evaluate the model

Split, Test, and Train



Hold Out Method

- Depending on each case, data can be split into 80–20, 75–25, 70–30, or even 50–50 in some cases.

- The larger set of data will be used as Training Set

- Pros:
  Good computational cost

- Cons:
  High Variance in results

Hold Out Method

```
data = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
from sklearn.model_selection import train_test_split
```

```
train, test = train_test_split(data, test_size=.2, random_state=1)
print("Train Data:", train, "Test Data:", test)
```

```
Train Data: [7, 5, 1, 4, 2, 8, 9, 6] Test Data: [3, 10]
```

k-Fold Cross Validation

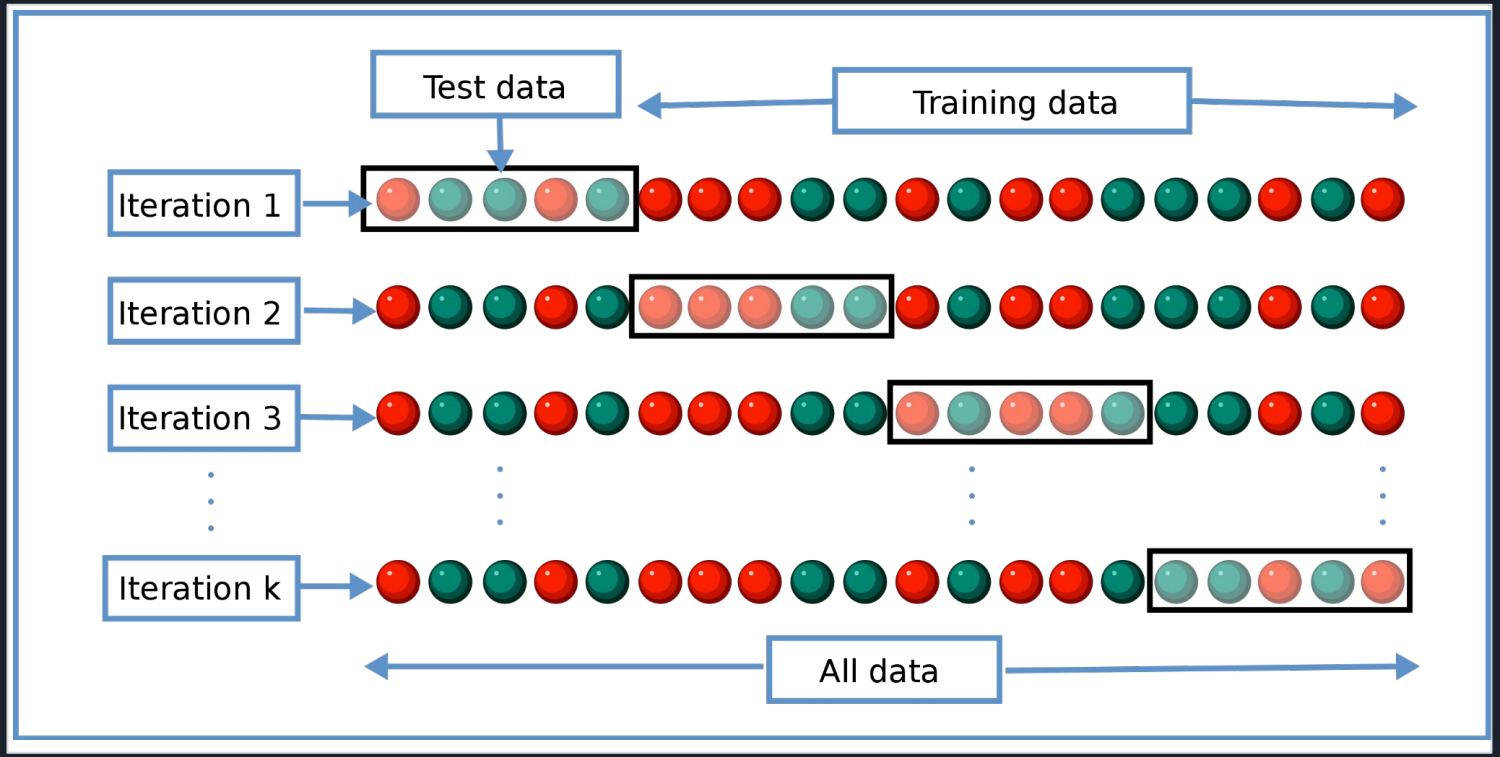
- Split the Data into k portions or blocks, and use the first blocks as Testing Data Set

- After the first round, iterate through other blocks to treat different block as Testing Data Set

- Pros:
  Worse computational cost compares to the Hold Out Method

- Cons:
  Lower Variance/Unbiased results compares to the Hold Out Method

k-Fold Cross Validation



k-Fold Cross Validation

```
from sklearn.model_selection import KFold
```

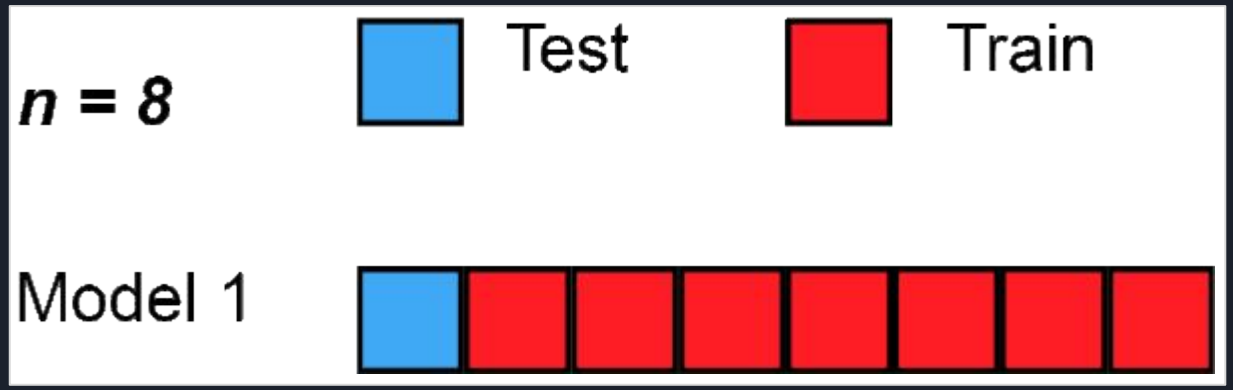
```
[ ] kf = KFold(n_splits=5, shuffle=False, random_state=None)

    for train, test in kf.split(data):
      print("Train Data:", train, "Test Data:", test)
```

```
Train Data: [2 3 4 5 6 7 8 9] Test Data: [0 1]
Train Data: [0 1 4 5 6 7 8 9] Test Data: [2 3]
Train Data: [0 1 2 3 6 7 8 9] Test Data: [4 5]
Train Data: [0 1 2 3 4 5 8 9] Test Data: [6 7]
Train Data: [0 1 2 3 4 5 6 7] Test Data: [8 9]
```

Leave One Out Cross Validation

- Extreme case for k-Fold CV, as each data point is count as a portion
- Resample through each data point and treat it as testing data
- Pros:
  Worst computational cost; iterate n times for n numbers of data
- Cons:
  Lowest Variance/Least Biased in outcome

Leave One Out Cross Validation



Leave One Out Cross Validation

```
from sklearn.model_selection import LeaveOneOut
```

```
[ ] l = LeaveOneOut()

    for train, test in l.split(data):
      print("Train Data:", train, "Test Data:", test)
```

```
Train Data: [1 2 3 4 5 6 7 8 9] Test Data: [0]
Train Data: [0 2 3 4 5 6 7 8 9] Test Data: [1]
Train Data: [0 1 3 4 5 6 7 8 9] Test Data: [2]
Train Data: [0 1 2 4 5 6 7 8 9] Test Data: [3]
Train Data: [0 1 2 3 5 6 7 8 9] Test Data: [4]
Train Data: [0 1 2 3 4 6 7 8 9] Test Data: [5]
Train Data: [0 1 2 3 4 5 7 8 9] Test Data: [6]
Train Data: [0 1 2 3 4 5 6 8 9] Test Data: [7]
Train Data: [0 1 2 3 4 5 6 7 9] Test Data: [8]
Train Data: [0 1 2 3 4 5 6 7 8] Test Data: [9]
```

Conclusion

- Cross-Validation is a very useful tool when comes to evaluating the Maching Learning method.

- Depends on the data set and use case, the data scientist can choose from using a method that is easy on computation or low on biased for the result.

Thank you for the participant!