Airfoils with Data Analytics

By Ruangyot Nanchiang

Simulation with JavaFoil

ก่อนที่เราจะนำข้อมูลของ Airfoil แต่ละอันที่เราได้ทำการเลือก มาทำการ simulation ใน JavaFoil เสียก่อน เนื่องจากว่าสามารถประมาณค่าต่างๆได้รวดเร็วกว่า Anasys ซึ่งเงื่อนไขในการทำ Simulation ใน JavaFoil มีดังต่อไปนี้

1. Re = 4000000
2. Mach Number = 0.3
3. Cd and Angle of Attack at Cl=0.5
4. Thickness at 0.75c

จากนั้นก็นำข้อมูลที่ได้จากการทำ Simulation มาเก็บไว้ในโปรแกรม Excel และ save เป็น .csv เพื่อนำไป import และวิเคราะห์ต่อโดยใช้ Python

Data Analytics with Machine Learning

ในขั้นตอนนี้เราจะนำข้อมูลที่ได้มาจากการทำ Simulation จาก JavaFoil มาดำเนินการร่วมกับ Machine Learning เพื่อแบ่ง Airfoils ออกเป็นกลุ่มต่างๆ โดยจะวิเคราะห์จากตัวแปรที่ได้ทำการ Simulation มาก่อนหน้า ได้แก่ Cd, Cl/Cd และ t0.75c

Model ที่ได้เลือกใช้คือ K means clustering algorithm เนื่องจาก K means clustering เป็น algorithm ที่ใช้ในการแบ่งแยกกลุ่มข้อมูลจากกลุ่มใหญ่ๆให้เป็นกลุ่มย่อย โดยจะทำการหาความสัมพันธ์จากตัวแปรในข้อมูลทั้งหมด และแบ่งกลุ่มออกมา การจะบอกได้ว่าแบ่งเป็นกี่กลุ่มจึงจะเหมาะสม เราจะพิจารณาจาก silhouette score ยิ่งเข้าใกล้ 1 มากเท่าไหร่ นั่นหมายความว่าการแบ่งกลุ่มของข้อมูลยิ่งมีประสิทธิภาพ (โดยก่อนเริ่มทำการใช้ K means clustering เราต้องมีการ transform ขนาดของข้อมูลก่อนเสมอ เพื่อให้ค่าของข้อมูลอยู่ในช่วงที่มีการกระจายอย่างเหมาะสม)

```
In [5]: df[cols].hist(layout=(1, len(cols)), figsize=(3*len(cols),2.5));
```

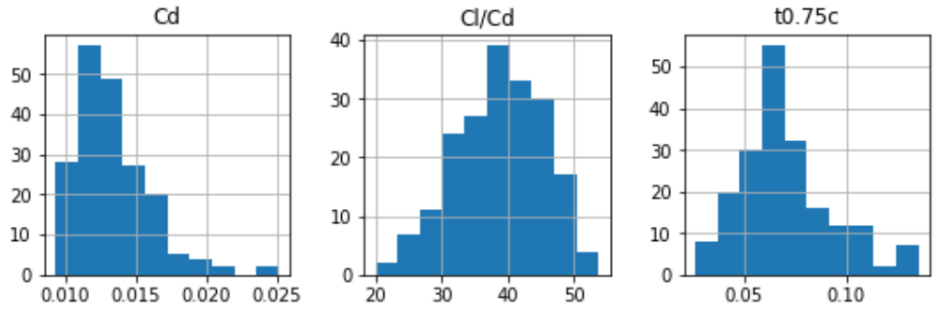


Figure 1 ก่อนทำการ transform ข้อมูล

```
In [8]: X[cols].hist(layout=(1, len(cols)), figsize=(3*len(cols),2.5), color='#ffcc00');
```

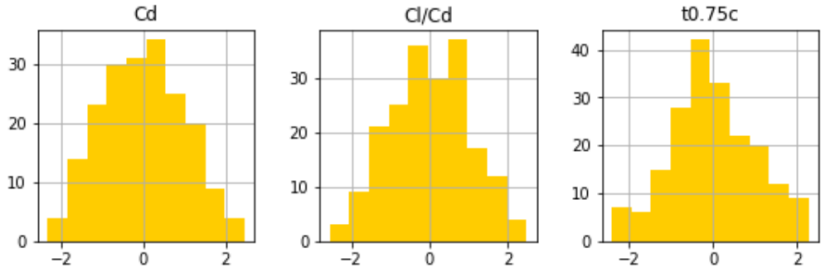


Figure 2 หลังทำการ transform ข้อมูล

จากข้อมูล Airfoils ทั้งหมดที่มี เราได้ทำการ coding ให้ Machine Learning ได้ทำการแบ่งข้อมูลออกมา โดยมี silhouette score ของการแบ่งดังนี้

In [10]:
```
def sil_score(X, from_k=2, to_k=6):
    '''
    calculate silhouette score for k clusters
    '''
    sils=[]
    for k in range(from_k, to_k + 1):
        m = KMeans(n_clusters=k)
        m.fit(X)
        # The silhouette_score gives the average value for all the samples
        silhouette_avg = silhouette_score(X, m.labels_).round(4)
        sils.append([silhouette_avg, k])

    return sils
```

This is function for calculate the number of cluster.

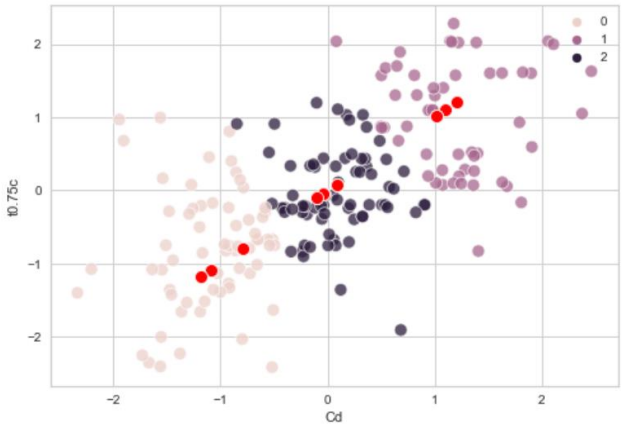
In [11]:
```
ss=sil_score(X, 2, 5)
print(f'scores = {ss}')
print(f'optimal number of clusters = {max(ss)[1]}')
```

```
scores = [[0.4772, 2], [0.3816, 3], [0.3514, 4], [0.3585, 5]]
optimal number of clusters = 2
```

จะสังเกตได้ว่า ถ้าหากเราแบ่งข้อมูลออกเป็น 2 กลุ่ม จะมี silhouette score สูงถึง 0.4772 แต่เนื่องจากว่าเราต้องการแบ่งออกเป็น 3 กลุ่ม เพื่อง่ายต่อการเปรียบเทียบระหว่าง t0.75c กับ Cl/Cd ซึ่งการแบ่งเป็น 3 กลุ่ม จะได้ silhouette score = 0.3816 รองจากการแบ่งข้อมูลออกเป็น 2 กลุ่ม

หลังจากที่เราได้ทำการแบ่งกลุ่มของข้อมูลทั้งหมดแล้ว Machine Learning จะทำการคำนวณหาจุด Mean ของตัวแปรในแต่ละจุดให้และแยกประเภทของข้อมูลให้ดังภาพ

```
In [32]: sns.scatterplot(x=X['Cd'], y=X['t0.75c'], s=100, hue=X['cluster'], alpha=0.75)
         for i in range(3):
             sns.scatterplot(x=point[i], y=point[i], s=100, color='red')
             sns.scatterplot(x=point[i], y=point[i], s=100, color='red')
             sns.scatterplot(x=point[i], y=point[i], s=100, color='red')
```




```
In [16]: point = model.cluster_centers_
         point

Out[16]: array([[-0.79522635, -1.09414976,  1.09683056],
                [ 1.01168643,  1.20138708, -1.1858663 ],
                [-0.04325773,  0.08407664, -0.09819412]])
```

Figure 3 พิกัดของการแบ่งกลุ่มข้อมูล

เมื่อทำการแบ่งข้อมูลเรียบร้อยแล้ว ก็ทำการ save เป็นไฟล์ใหม่ เพื่อไปทำการเลือกต่อว่า Airfoils อันไหนจะมีประสิทธิภาพดีที่สุดและหนาที่สุด

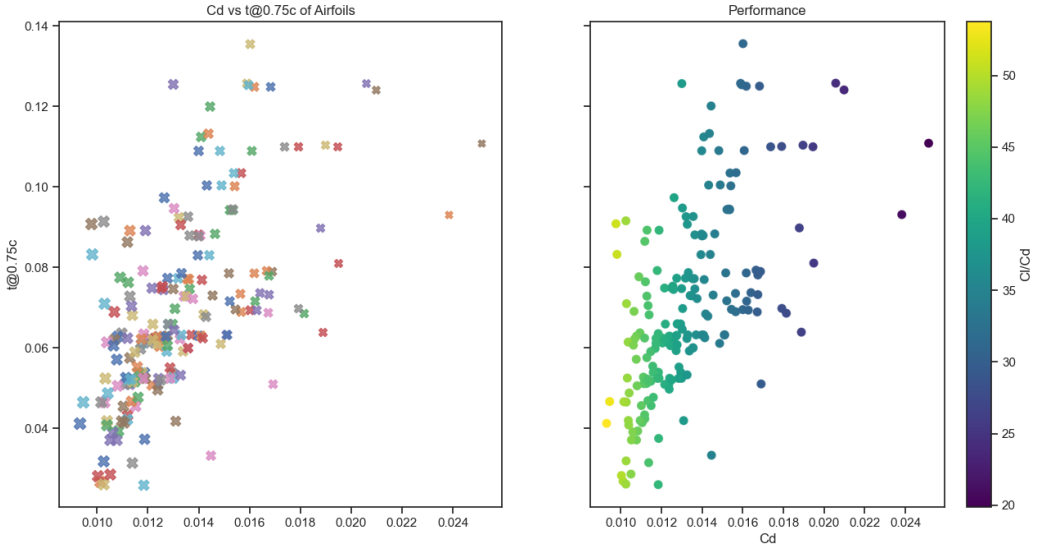
	Airfoils	t0.75c	Cd	Cl/Cd	cluster
3	NACA 63-210	0.04332	0.01117	44.762757	0
4	NACA 63-212	0.05112	0.01127	44.365572	0
7	NACA 63-412	0.05108	0.01113	44.923630	0
20	NACA 66-021	0.12502	0.01682	29.726516	1
29	NACA 63(4)-221	0.08318	0.01440	34.722222	1
42	NACA 64(4)-221	0.08830	0.01462	34.199726	1
0	NACA 63A010	0.05090	0.01236	40.453074	2
1	NACA 63012A	0.06052	0.01244	40.192926	2
2	NACA 63-015A	0.07462	0.01364	36.656892	2

Figure 4 ตัวอย่างของข้อมูลที่ผ่านการแบ่งโดย K means clustering แล้ว

Discussion Part

ในขั้นตอนนี้เราจะทำการตัดสินใจเลือกว่า Airfoils อันไหนที่มีประสิทธิภาพสูงที่สุด และมีความหนาที่ 0.75c มากที่สุด จากกลุ่มข้อมูลของ Airfoils ที่แบ่งมาแล้วจากการใช้ Machine Learning

ก่อนอื่น เราต้องทำการ visualizer ข้อมูลโดยรวมก่อน เพื่อประกอบการตัดสินใจ และตรวจสอบความถูกต้องของข้อมูลเทียบจากหลักความเป็นจริงที่ควรจะเป็นได้



จากข้อมูลดังกล่าว ค่อนข้างมีความถูกต้อง เพราะว่าเป็นไปตามหลักความเป็นจริงคือ ถ้า Airfoils ยิ่งหนา Cd ก็จะยิ่งเยอะ และ Cl/Cd ก็จะน้อยลง

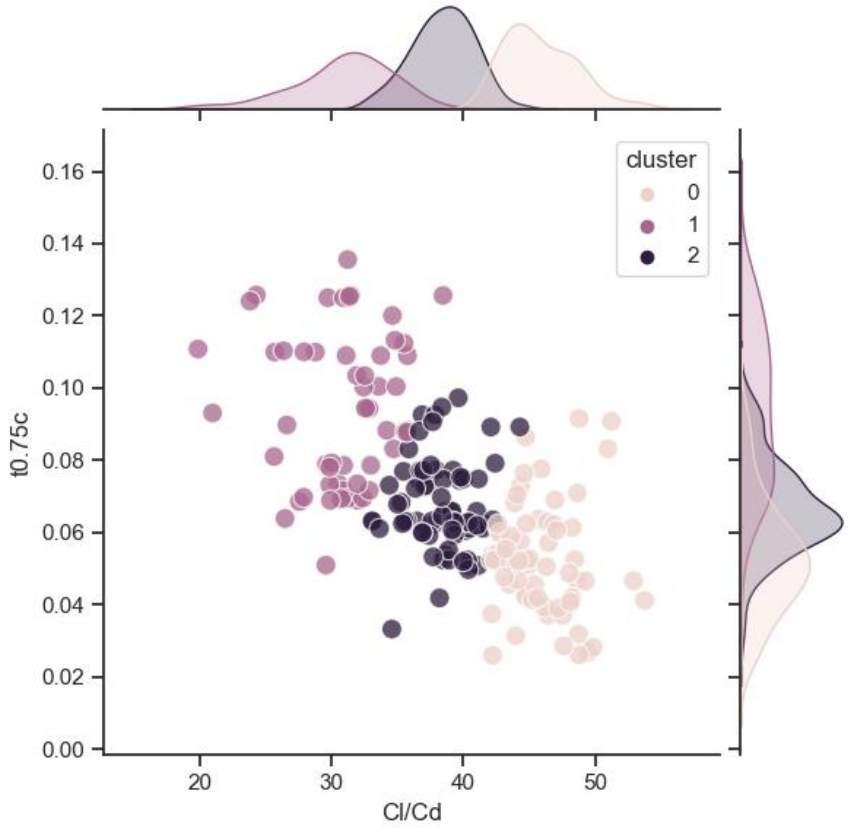
Calculate statistics

```
In [3]: df.describe()
```

Out[3]:

	t0.75c	Cd	Cl/Cd	cluster
count	194.000000	194.000000	194.000000	194.000000
mean	0.069950	0.013306	38.808295	1.036082
std	0.022935	0.002583	6.580826	0.847879
min	0.025972	0.009300	19.888624	0.000000
25%	0.053413	0.011362	34.435458	0.000000
50%	0.065243	0.012780	39.123631	1.000000
75%	0.080600	0.014520	44.004407	2.000000
max	0.135580	0.025140	53.763441	2.000000

จากนั้น เราจะนำข้อมูลที่ได้ทำการแบ่งแล้วมาเลือก โดยกลุ่มที่เลือกจะเป็นกลุ่มที่ 0 เพราะเป็นกลุ่มที่มี Cl/Cd มากที่สุด



หลังจากที่ตัดสินใจได้แล้วว่าจะเลือกกลุ่มข้อมูลที่ 0 เราก็ต้องมาตั้งสมมุติฐานอีกทีว่ามันมีความหนาไหม ถ้าเทียบกับ ความหนาเฉลี่ยของข้อมูลกลุ่มที่ 1 ซึ่งเป็นกลุ่มที่มีความหนามากที่สุด โดยเราจะคัดเลือก Airfoils ในกลุ่มที่ 0 มา 5 ข้อมูล โดย sort จาก 0.75c และ Cl/Cd และนำไปผ่านกระบวนการ Hypothesis Test เพื่อให้แน่ใจว่า Airfoils กลุ่มนี้มีความหนาหรือไม่

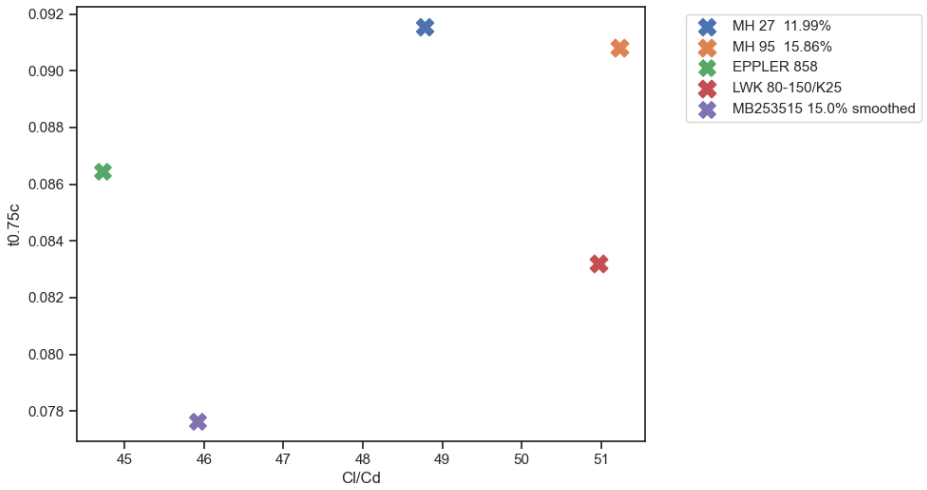


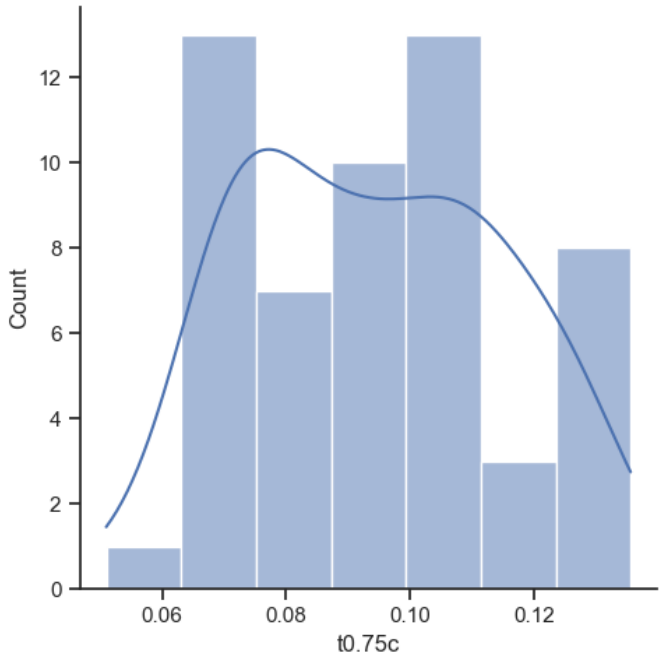
Figure 5 Airfoils กลุ่มที่มีคัดแล้ว

	t0.75c	Cd	Cl/Cd	cluster
count	5.000000	5.000000	5.000000	5.0
mean	0.085928	0.010378	48.322959	0.0
std	0.005740	0.000638	2.933516	0.0
min	0.077640	0.009760	44.722719	0.0
25%	0.083200	0.009810	45.913682	0.0
50%	0.086440	0.010250	48.780488	0.0
75%	0.090812	0.010890	50.968400	0.0
max	0.091546	0.011180	51.229508	0.0

Figure 6 Statistics ของกลุ่ม Airfoil ที่ผ่านการ sort แล้ว

เนื่องจากข้อมูลกลุ่มที่ 1 มีการกระจายความหนาในช่วง mean ประมาณ 40% เราจึงจะกำหนดช่วงความเชื่อมั่นในการทำ Hypothesis test 0.4 ดังนั้น จะได้ค่า Alpha = 0.6 แบ่งออกเป็น Two way test จะได้ว่า Alpha = 0.3

***หมายเหตุ ที่ต้องการ Test เป็น Two Ways เพราะว่า ต้องการทราบว่า Airfoils กลุ่มนี้มีความหนาเทียบเคียงกับ Airfoils กลุ่มที่มีความหนามากที่สุดได้หรือไม่



	t0.75c	Cd	Cl/Cd	cluster
count	55.000000	55.000000	55.000000	55.0
mean	0.094136	0.016465	30.878222	1.0
std	0.020716	0.002334	3.746153	0.0
min	0.051000	0.013000	19.888624	1.0
25%	0.075886	0.015185	29.612105	1.0
50%	0.094310	0.016000	31.250000	1.0
75%	0.109990	0.016885	32.927320	1.0
max	0.135580	0.025140	38.461538	1.0

Figure 7 Statistics ของ Airfoils กลุ่มที่ 1

In [17]:
```
# Define alpha at 0.6
# Two ways test, Thus alpha was devide by 2. alhpa = 0.3 for each side.
# H0=t.mean(), H1 != t.mean()

Z = ((st_sort['t0.75c'].mean()-t['t0.75c'].mean())/(st_sort['t0.75c'].mean()*np.sqrt(st_sort['t0.75c'].

if -0.52 < Z < 0.52: # @alpha=0.3 Z = 0.52
    print(f'Z = {Z}')
    print('Accept H0 @alpha=0.6: Airfoils are thick')
else:
    print(f'Z = {Z}')
    print('Reject H0 @alpha=0.6: Airfoils are thin')
```

```
Z = -0.042723
Accept H0 @alpha=0.6: Airfoils are thick
```

หลังจากที่ได้ทำการ Hypothesis Test แล้ว เราสามารถบอกได้ว่า Airfoils กลุ่มที่เลือกมานี้มีความหนาเทียบเคียงได้กับ กลุ่ม Airfoils ที่มีความหนามากที่สุด