

LAB 1: BÀI THỰC HÀNH THAO TÁC DỮ LIỆU

Câu 1: Tải dữ liệu từ file CSV

```
1 import pandas as pd
2
3 # Câu 1: Tải dữ liệu từ file CSV
4 data = pd.read_csv('dulieuxettuyendaihoc.csv')
```

Câu 2: Phân loại dữ liệu định tính và định lượng

```
1 qualitative_columns = ['GT', 'DT', 'KV', 'KT']
2 quantitative_columns = [col for col in data.columns if col not in qualitative_columns + ['STT']]
```

Câu 3: In ra 10 dòng đầu và 10 dòng cuối

```
1 print("\nCâu 3:")
2 print("10 dòng đầu:")
3 print(data.head(10))
4 print("\n10 dòng cuối:")
5 print(data.iloc[-10:])
```

```
Câu 3:
10 dòng đầu:
   STT  T1  L1  H1  S1  V1  X1  D1  N1  T2  L2  H2  S2  V2  X2  D2  N2  T3  ...  X5  D5  N5  T6  L6  H6  S6  V6  X6  D6  N6  GT  DT  KV  DH1  DH2  DH
0  1  7.2  7.3  6.3  7.3  7.0  7.9  7.3  5.5  8.4  8.9  5.8  9.0  6.6  8.7  7.2  5.8  7.4  ...  5.0  7.4  5.0  6.9  5.7  6.5  7.2  8.4  6.6  7.6  5.9  F  NaN  2NT  3.25  3.25  4.5
1  2  5.4  3.9  3.9  4.0  5.4  5.4  5.3  2.8  6.3  5.4  3.2  7.2  5.4  6.1  4.7  4.7  4.3  ...  6.3  6.7  4.4  4.0  5.4  4.0  5.2  5.5  6.6  6.1  4.4  M  NaN  1  6.00  4.00  3.5
2  3  5.6  6.8  7.2  7.5  4.3  7.4  5.8  3.2  5.0  4.9  6.2  6.1  5.3  7.1  4.0  4.9  2.8  ...  6.9  3.9  3.2  5.7  6.6  6.8  6.8  4.1  7.9  8.1  4.6  M  NaN  1  5.00  6.75  4.0
3  4  6.6  6.4  5.3  6.9  5.4  7.3  6.4  5.8  5.1  6.2  5.5  5.5  4.4  6.5  4.9  4.4  5.9  ...  5.5  5.9  6.5  7.4  5.6  6.6  6.8  6.6  7.1  7.3  7.4  M  NaN  1  4.25  4.25  5.2
4  5  6.0  5.0  6.0  7.3  6.5  7.7  7.9  6.1  5.4  5.5  5.9  7.5  6.5  6.9  7.6  6.4  7.6  ...  7.0  7.0  6.3  8.0  6.2  6.8  6.9  6.6  6.1  7.5  7.2  M  NaN  2NT  4.25  4.50  5.0
5  6  9.3  7.6  7.9  8.6  7.0  7.3  7.7  7.9  9.6  7.8  8.0  8.7  7.7  8.0  9.2  9.1  7.1  ...  6.6  6.6  8.2  7.8  6.4  7.2  7.7  7.7  5.7  8.0  7.8  M  NaN  1  1.50  4.00  6.0
6  7  2.8  3.9  5.5  6.9  5.0  7.3  4.6  5.2  4.4  6.7  5.5  7.7  5.0  6.6  6.1  4.6  4.6  ...  6.1  6.1  4.8  5.3  7.0  6.0  5.7  6.1  6.6  6.0  6.0  F  NaN  2  6.50  6.75  5.2
7  8  8.3  6.0  7.6  5.1  7.5  4.7  5.8  7.2  6.7  6.3  6.4  7.1  7.3  6.9  6.1  7.9  8.6  ...  4.8  7.4  6.9  8.3  6.7  5.6  5.9  7.3  7.1  6.8  7.0  F  NaN  2  3.75  4.50  4.2
8  9  6.5  6.3  7.6  6.0  5.5  7.1  6.3  5.0  7.3  6.8  7.3  6.5  6.4  7.9  7.6  3.7  6.4  ...  5.6  7.1  6.7  7.7  8.7  8.9  7.5  6.0  9.1  7.9  6.1  F  NaN  1  3.50  3.50  6.7
9 10  7.3  5.9  4.7  7.1  6.7  7.9  6.7  7.7  8.0  6.8  7.6  6.9  6.8  7.8  6.4  7.4  6.5  ...  6.3  5.1  7.1  6.9  7.0  6.7  7.1  7.7  6.4  6.1  7.8  F  NaN  1  4.00  4.75  5.5
[10 rows x 56 columns]
```

10 dòng cuối:

	STT	T1	L1	H1	S1	V1	X1	D1	N1	T2	L2	H2	S2	V2	X2	D2	N2	T3	...	X5	D5	N5	T6	L6	H6	S6	V6	X6	D6	N6	GT	DT	KV	DH1	DH2	D
H3	KT																		...																	
90	A	8.1	7.7	9.1	8.5	6.1	8.6	8.8	7.3	8.8	8.3	8.4	9.3	7.1	8.9	8.8	7.2	9.2	...	7.6	8.1	7.6	8.5	9.2	7.5	8.6	7.9	8.3	8.7	7.8	M	NaN	2	6.25	4.00	6.
91	A	7.8	6.5	6.7	5.4	6.2	4.8	5.9	4.4	8.9	6.9	6.2	7.5	6.2	4.4	6.4	4.7	8.5	...	6.4	6.4	5.5	9.4	8.2	7.1	8.2	6.3	7.2	7.4	7.0	M	NaN	2NT	4.75	4.75	4.
92	A	5.0	6.6	6.5	7.2	5.8	6.7	6.4	6.0	6.8	6.1	6.5	8.6	5.9	4.9	6.3	7.5	7.2	...	5.6	5.4	5.6	7.5	6.3	6.7	6.8	7.2	5.9	6.7	5.8	M	NaN	1	3.25	5.25	4.
93	A	5.2	5.2	6.8	7.9	6.6	8.9	7.6	5.3	6.8	5.8	5.8	8.9	7.0	8.8	8.6	5.7	6.7	...	7.4	7.5	5.4	7.3	6.5	6.7	5.8	7.3	8.7	7.8	5.0	M	NaN	1	3.50	4.25	5.
94	A	5.8	5.9	7.6	6.1	5.3	8.1	6.1	5.0	6.4	7.2	7.7	7.5	5.3	8.4	6.8	6.1	7.3	...	5.3	7.4	6.5	6.7	8.1	7.1	6.8	6.0	7.3	6.8	5.3	M	NaN	1	4.25	2.50	4.
95	A	8.6	6.9	7.4	8.8	7.6	5.8	7.3	5.7	8.9	6.8	6.5	7.3	7.0	7.3	6.9	7.6	7.7	...	5.3	5.2	5.2	6.6	6.9	7.4	7.0	7.6	6.3	6.1	6.2	F	NaN	1	5.25	1.50	6.
96	C	3.7	5.4	6.0	5.1	5.5	3.9	6.1	4.4	4.1	4.8	5.1	4.7	5.9	6.9	4.6	5.4	3.1	...	7.6	7.4	4.9	5.7	7.7	5.0	7.0	6.5	7.9	7.5	4.4	F	NaN	1	5.25	3.75	4.
97	C	8.8	5.5	7.4	7.7	6.2	7.3	8.1	4.5	9.5	7.3	6.0	8.4	7.0	8.4	8.8	6.5	9.5	...	8.6	8.6	5.9	8.5	7.1	6.2	8.6	7.0	9.6	8.4	5.8	M	NaN	2NT	7.00	8.00	4.
98	C	2.7	1.8	3.4	5.3	4.5	7.9	4.9	3.8	2.8	5.7	5.1	5.0	5.5	6.6	3.9	4.3	6.2	...	5.9	5.3	4.5	4.3	5.6	6.3	4.7	5.0	6.6	5.2	5.9	M	NaN	1	5.00	3.50	5.
99	C	4.1	5.2	4.9	5.3	5.5	5.4	7.2	5.4	4.4	6.4	4.8	7.2	6.0	6.3	6.8	4.9	6.0	...	7.9	6.9	4.7	5.1	6.4	5.9	5.4	6.3	5.6	6.6	5.8	M	NaN	2NT	5.25	2.50	4.
100	C																		...																	

[10 rows x 56 columns]

Câu 4: Thống kê dữ liệu thiếu cho cột DT và thay thế dữ liệu thiếu bằng 0

```

1 print("\nCâu 4:")
2 print("\nThống kê dữ liệu thiếu cho cột DT:")
3 print(data['DT'].value_counts(dropna=False))
4 data['DT'] = data['DT'].fillna(0)
5 print("\nSau khi thay thế dữ liệu thiếu bằng 0:")
6 print(data['DT'].value_counts())

```

Câu 4:

Thống kê dữ liệu thiếu cho cột DT:

```

DT
NaN      97
1.0       2
6.0       1
Name: count, dtype: int64

```

Sau khi thay thế dữ liệu thiếu bằng 0:

```

DT
0.0      97
1.0       2
6.0       1
Name: count, dtype: int64

```

Câu 5: Thống kê dữ liệu thiếu cho T1 và thay thế bằng giá trị trung bình

```

1 print("\nCâu 5:")
2 print("\nThống kê dữ liệu thiếu cho cột T1 trước khi xử lý:")
3 print(data['T1'].isnull().sum())
4 mean_T1 = data['T1'].mean()
5 data['T1'] = data['T1'].fillna(mean_T1)
6 print("\nSau khi thay thế dữ liệu thiếu bằng giá trị trung bình:")
7 print(data['T1'].isnull().sum())

```

Câu 5:

Thống kê dữ liệu thiếu cho cột T1 trước khi xử lý:
0

Sau khi thay thế dữ liệu thiếu bằng giá trị trung bình:
0

Câu 6: Xử lý dữ liệu thiếu cho các cột điểm số còn lại bằng giá trị trung bình



```
1 print("\nCâu 6:")
2 for col in quantitative_columns:
3     if data[col].isnull().sum() > 0:
4         mean_value = data[col].mean()
5         data[col].fillna(mean_value, inplace=True)
```

Câu 7: Tạo các biến TBM1, TBM2, TBM3



```
1 print("\nCâu 7:")
2 data['TBM1'] = (data['T1']*2 + data['L1'] + data['H1'] + data['S1'] + data['V1']*2 + data['X1'] + data['D1'] + data['N1']) / 10
3 data['TBM2'] = (data['T2']*2 + data['L2'] + data['H2'] + data['S2'] + data['V2']*2 + data['X2'] + data['D2'] + data['N2']) / 10
4 data['TBM3'] = (data['T6']*2 + data['L6'] + data['H6'] + data['S6'] + data['V6']*2 + data['X6'] + data['D6'] + data['N6']) / 10
5 print(data[['TBM1', 'TBM2', 'TBM3']])
```

Câu 7:

	TBM1	TBM2	TBM3
0	7.00	7.54	7.01
1	4.69	5.47	5.07
2	5.77	5.38	6.04
3	6.21	5.20	6.88
4	6.50	6.36	6.99
..
95	7.43	7.42	6.83
96	4.93	5.15	6.39
97	7.05	7.84	7.67
98	4.15	4.72	5.29
99	5.26	5.72	5.85

[100 rows x 3 columns]

Câu 8: Tạo các biến xếp loại XL1, XL2, XL3

```
1 print("\nCâu 8:")
2 def classify_grade(tbm):
3     if tbm < 5.0:
4         return 'Y'
5     elif 5.0 <= tbm < 6.5:
6         return 'TB'
7     elif 6.5 <= tbm < 8.0:
8         return 'K'
9     elif 8.0 <= tbm < 9.0:
10        return 'G'
11    else:
12        return 'XS'
13
14 data['XL1'] = data['TBM1'].apply(classify_grade)
15 data['XL2'] = data['TBM2'].apply(classify_grade)
16 data['XL3'] = data['TBM3'].apply(classify_grade)
17
18 print(data[['XL1', 'XL2', 'XL3']])
```

Câu 8:

	XL1	XL2	XL3
0	K	K	K
1	Y	TB	TB
2	TB	TB	TB
3	TB	TB	K
4	K	TB	K
..
95	K	K	K
96	Y	TB	TB
97	K	K	K
98	Y	Y	TB
99	TB	TB	TB

[100 rows x 3 columns]

Câu 9: Tạo các biến US_TBM1, US_TBM2, US_TBM3 theo thang điểm 4 của Mỹ

```
1 print("\nCâu 9:")
2 def min_max_normalization(value, min_val=0, max_val=10, target_min=0, target_max=4):
3     return (value - min_val) / (max_val - min_val) * (target_max - target_min) + target_min
4
5 data['US_TBM1'] = data['TBM1'].apply(min_max_normalization)
6 data['US_TBM2'] = data['TBM2'].apply(min_max_normalization)
7 data['US_TBM3'] = data['TBM3'].apply(min_max_normalization)
8
9 print(data[['US_TBM1', 'US_TBM2', 'US_TBM3']])
```

Câu 9:

	US_TBM1	US_TBM2	US_TBM3
0	2.800	3.016	2.804
1	1.876	2.188	2.028
2	2.308	2.152	2.416
3	2.484	2.080	2.752
4	2.600	2.544	2.796
..
95	2.972	2.968	2.732
96	1.972	2.060	2.556
97	2.820	3.136	3.068
98	1.660	1.888	2.116
99	2.104	2.288	2.340

[100 rows x 3 columns]

Câu 10: Tạo biến KQXT để xác định sinh viên đậu hoặc rớt

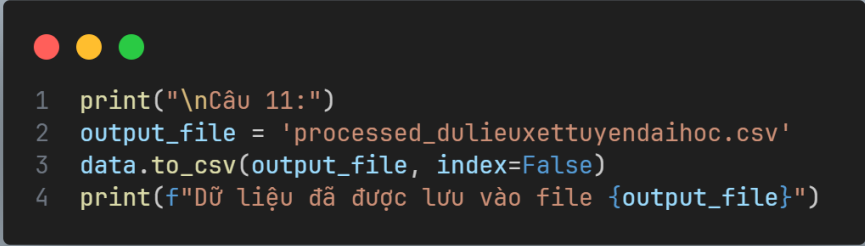
```
1 print("\nCâu 10:")
2 def determine_admission_result(dh1, dh2, dh3, kt):
3     if kt in ['A', 'A1']:
4         score = (dh1 * 2 + dh2 + dh3) / 4
5     elif kt == 'B':
6         score = (dh1 + dh2 * 2 + dh3) / 4
7     else:
8         score = (dh1 + dh2 + dh3) / 3
9     return 1 if score >= 5.0 else 0
10
11 data['KQXT'] = data.apply(lambda row: determine_admission_result(row['DH1'], row['DH2'], row['DH3'], row['KT']), axis=1)
12
13 print(data['KQXT'])
```

Câu 10:

0	0
1	0
2	1
3	0
4	0
..	
95	0
96	0
97	1
98	0
99	0

Name: KQXT, Length: 100, dtype: int64

Câu 11: Lưu dữ liệu đã xử lý xuống file CSV



```
1 print("\nCâu 11:")
2 output_file = 'processed_dulieuxettuyendaihoc.csv'
3 data.to_csv(output_file, index=False)
4 print(f"Dữ liệu đã được lưu vào file {output_file}")
```

Câu 11:

Dữ liệu đã được lưu vào file chaugiakiet_dulieuxettuyendaihoc.csv