SC310005 Artificial Intelligence

Lecture 2: Basic Pandas (Part I and II)

teerapong.pa@chula.ac.th

Week 2: Basic Pandas II

Loading the dataset: Read the Titanic dataset using Pandas.

[1]
```
import pandas as pd

# Load the Titanic dataset
titanic_data = pd.read_csv('https://raw.githubusercontent.com/kaopanboonyuen/SC310005_ArtificialIntelligence_2023s1/main/dataset/titanic_dataset.csv')
```

Basic Function

[2]
```
# Viewing the first few rows: Use head() to view the initial rows.

titanic_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

[3]
```
# Checking data types: Use dtypes to check the data types of columns.

titanic_data.dtypes
```

```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

[4]
```
# Summary statistics: Obtain summary statistics of numerical columns.

titanic_data.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

[5]
```
# Data Shape

titanic_data.shape
```

```
(891, 12)
```

Counting unique values in a column: Use value_counts() to count unique values in a column.

```
[6]  titanic_data['Sex'].value_counts()
```

```
male      577
female    314
Name: Sex, dtype: int64
```

Grouping data: Group data based on a specific column.

```
[7]  # Mean age of passengers in each class:

     titanic_data.groupby('Pclass')['Age'].mean()
```

```
Pclass
1    38.233441
2    29.877630
3    25.140620
Name: Age, dtype: float64
```

```
[8]  # Total number of passengers in each class:

     titanic_data.groupby('Pclass').size()
```

```
Pclass
1    216
2    184
3    491
dtype: int64
```

```
[9]  # Maximum fare paid by passengers in each class:

     titanic_data.groupby('Pclass')['Fare'].max()
```

```
Pclass
1    512.3292
2     73.5000
3     69.5500
Name: Fare, dtype: float64
```

```
[10] # Count of survived passengers in each class:

     titanic_data.groupby('Pclass')['Survived'].sum()
```

```
Pclass
1    136
2     87
3    119
Name: Survived, dtype: int64
```

[11]
```
# Median age of male and female passengers in each class:

titanic_data.groupby(['Pclass', 'Sex'])['Age'].median()
```

```
Pclass  Sex
1       female    35.0
        male      40.0
2       female    28.0
        male      30.0
3       female    21.5
        male      25.0
Name: Age, dtype: float64
```

[12]
```
# Percentage of survived passengers in each class:

titanic_data.groupby('Pclass')['Survived'].mean() * 100
```

```
Pclass
1    62.962963
2    47.282609
3    24.236253
Name: Survived, dtype: float64
```

[13]
```
# Aggregating multiple columns using custom functions:

titanic_data.groupby('Pclass').agg({'Age': 'mean', 'Fare': 'max', 'Survived': 'sum'})
```

Pclass	Age	Fare	Survived
1	38.233441	512.3292	136
2	29.877630	73.5000	87
3	25.140620	69.5500	119

Conditional selection: Selecting rows based on a condition.

```
[15] female_passengers = titanic_data[titanic_data['Sex'] == 'female']
```

Applying a function to a column: Use apply() to transform a column.

```
[16] def age_category(age):
         if age < 18:
             return 'Child'
         else:
             return 'Adult'

     titanic_data['Age_Category'] = titanic_data['Age'].apply(age_category)
```

Creating a new variable: Create a new column based on existing columns.

```
[17] titanic_data['Family_Size'] = titanic_data['SibSp'] + titanic_data['Parch'] + 1
```

Filtering with two conditions: Select rows satisfying multiple conditions.

```
[18] survived_female_passengers = titanic_data[(titanic_data['Sex'] == 'female') & (titanic_data['Survived'] == 1)]
```

Null value handling: Check for missing values in the dataset.

```
[19] titanic_data.isnull().sum()
```

```
PassengerId       0
Survived          0
Pclass            0
Name              0
Sex               0
Age             177
SibSp             0
Parch             0
Ticket            0
Fare              0
Cabin           687
Embarked          2
Age_Category      0
Family_Size       0
dtype: int64
```

Filling missing values: Fill missing values in a column.

```
[20] titanic_data['Age'].fillna(titanic_data['Age'].median()) #  inplace=True
```

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    28.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

Dropping columns: Remove unnecessary columns from the dataset.

```
[21] titanic_data.drop(['Cabin', 'Ticket'], axis=1) # inplace=True
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked	Age_Category	Family_Size
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	Adult	2
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	Adult	2
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	Adult	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	Adult	2
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	Adult	1
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	13.0000	S	Adult	1
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	30.0000	S	Adult	1
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	23.4500	S	Adult	4
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	30.0000	C	Adult	1
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	7.7500	Q	Adult	1

891 rows × 12 columns

Sorting values: Sort the dataset based on a column.

```
[22] titanic_data.sort_values(by='Age', ascending=False)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_Category	Family_Size
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	30.0000	A23	S	Adult	1
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7750	NaN	S	Adult	1
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609	49.5042	NaN	C	Adult	1
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	A5	C	Adult	1
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7500	NaN	Q	Adult	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2292	NaN	C	Adult	1
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S	Adult	11
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN	S	Adult	1
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN	S	Adult	1
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S	Adult	4

891 rows × 14 columns

Exporting data: Save the modified dataset to a CSV file.

0s

```
titanic_data.to_csv('modified_titanic.csv', index=False)
```