Data Exploration
- File Handling

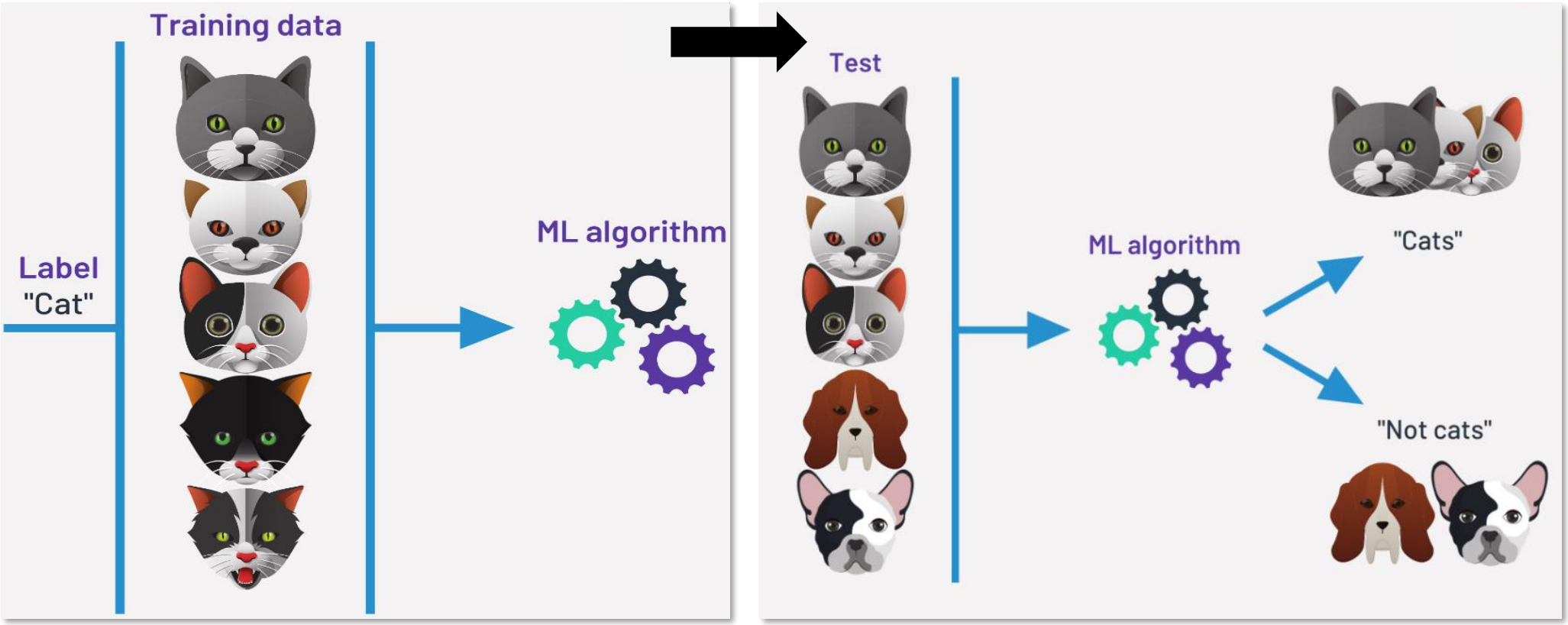
Dept. of Mechanical System Design Engineering,
Seoul National University of Science and Technology

Prof. Ju Yeon Lee
(jylee@seoultech.ac.kr)

Review

Data Analysis – Supervised Learning

- **Supervised Learning**
  machine learning approach defined by its use of labeled datasets to train algorithms to classify data and predict outcomes



Source: https://www.g2.com/articles/supervised-vs-unsupervised-learning

Data Analysis – Supervised Learning

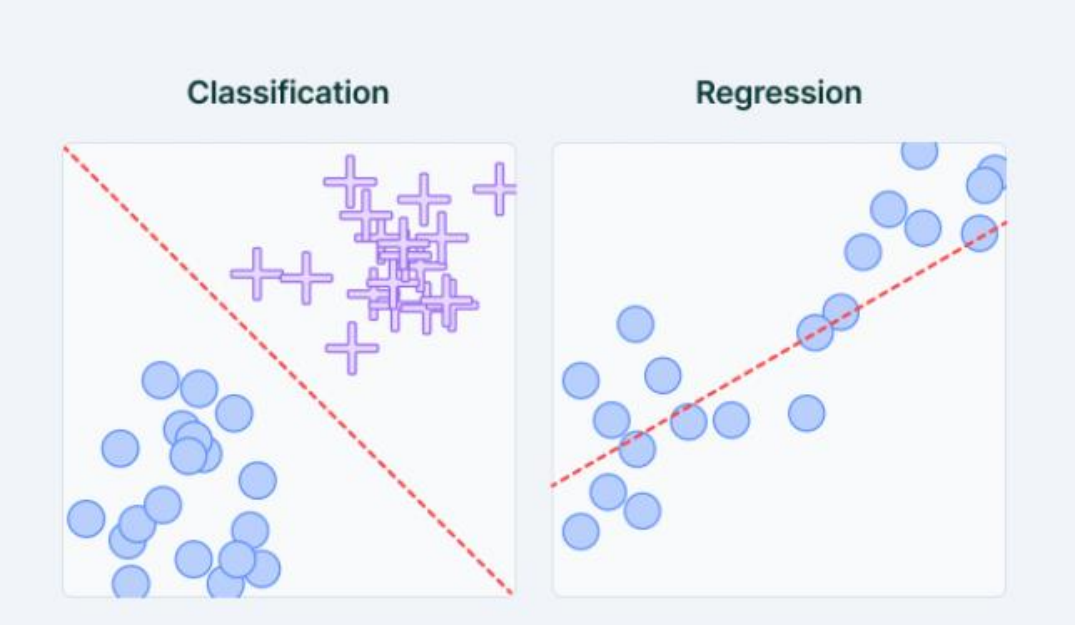
- **Supervised Learning**
  machine learning approach defined by its use of labeled datasets to train algorithms to classify data and predict outcomes

y-categorical value

Classification :

taking an input value and mapping it to a discrete value

typically consists of classes or categories



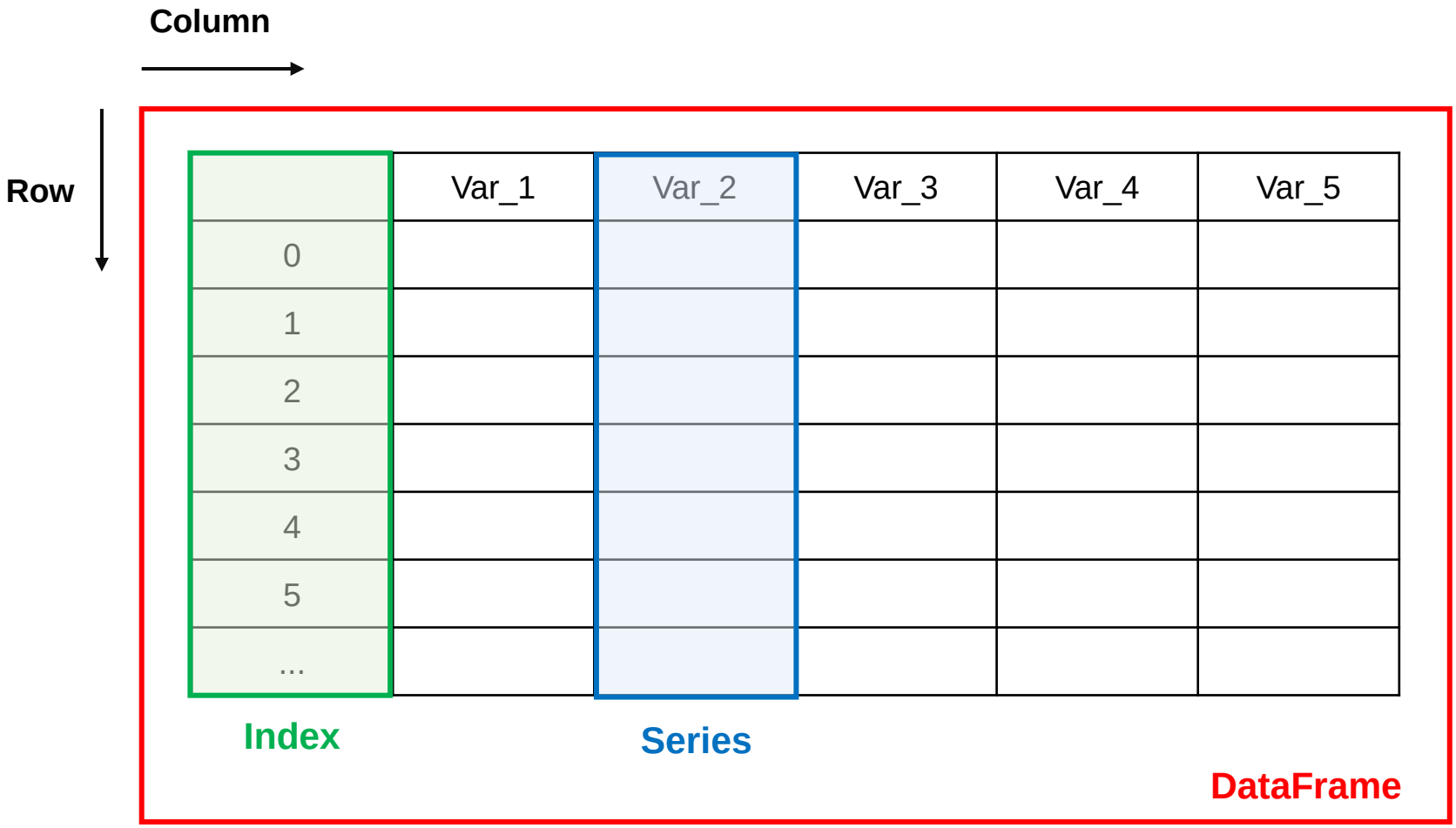
y-numerical value

Regression :

continuous data (value functions)

the predicted output values : real numbers

DataFrame


Column
Row
Var_1
Var_2
Var_3
Var_4
Var_5
0
1
2
3
4
5
...
Index
Series
DataFrame

DataFrame indexing & Slicing

```
# Indexing & Slicing DataFrame
data = [['S1', 25, 95],
        ['S2', 28, 80],
        ['S3', 22, 75]]
df=pd.DataFrame(data,
                index=['row1', 'row2', 'row3'],
                columns=['Name', 'Age', 'Score'])

# column observation
df['Name']
df['Age']
df['Score']
df[['Name', 'Score']]

# row observation
df['row1']          # Error
df.loc['row1']
df.loc[['row1', 'row3']]
```

```
# row & column observation
df.loc['row1', 'Name']
df.loc[:, 'Name']
df.loc[:, ['Name', 'Score']]
df.loc[:,'Name':'Score']

df.iloc[0,0]
df.iloc[:, [0,2]]
df.iloc[::2, [0,2]]
df.iloc[-1,:]
df.iloc[-1::-1,:]

# head and tail functions
df.head(1)
df.head(2)
df.tail(1)
df.tail(2)
```

Import/Export Data Files

File Handling (Pandas)

Format Type	Data Description	Reader	Writer
text	CSV	read_csv	to_csv
text	Fixed-Width Text File	read_fwf	
text	JSON	read_json	to_json
text	HTML	read_html	to_html
text	LaTeX		Styler.to_latex
text	XML	read_xml	to_xml
text	Local clipboard	read_clipboard	to_clipboard
binary	MS Excel	read_excel	to_excel
binary	OpenDocument	read_excel	

File Handling (Pandas)

binary	HDF5 Format	read_hdf	to_hdf
binary	Feather Format	read_feather	to_feather
binary	Parquet Format	read_parquet	to_parquet
binary	ORC Format	read_orc	to_orc
binary	Stata	read_stata	to_stata
binary	SAS	read_sas	
binary	SPSS	read_spss	
binary	Python Pickle Format	read_pickle	to_pickle
SQL	SQL	read_sql	to_sql
SQL	Google BigQuery	read_gbq	to_gbq

CSV File

- CSV (Comma-Separated Values) : a text file that uses a comma to separate values
- Separate columns with commas
- Separate rows by line break

Import CSV File → DataFrame : pandas.read_csv("file path (file name)")

pandas.read_csv

```
pandas.read_csv(filepath_or_buffer, sep=_NoDefault.no_default, delimiter=None,
header='infer', names=_NoDefault.no_default, index_col=None, usecols=None,
squeeze=None, prefix=_NoDefault.no_default, mangle_dupe_cols=True, dtype=None,
engine=None, converters=None, true_values=None, false_values=None,
skipinitialspace=False, skiprows=None, skipfooter=0, nrows=None, na_values=None,
keep_default_na=True, na_filter=True, verbose=False, skip_blank_lines=True,
parse_dates=None, infer_datetime_format=False, keep_date_col=False,
date_parser=None, dayfirst=False, cache_dates=True, iterator=False,
chunksize=None, compression='infer', thousands=None, decimal='.',
lineterminator=None, quotechar='"', quoting=0, doublequote=True, escapechar=None,
comment=None, encoding=None, encoding_errors='strict', dialect=None,
error_bad_lines=None, warn_bad_lines=None, on_bad_lines=None,
delim_whitespace=False, low_memory=True, memory_map=False, float_precision=None,
storage_options=None)                                                    [source]
```

CSV File

- CSV (Comma-Separated Values) : a text file that uses a comma to separate values
- Separate columns with commas
- Separate rows by line break

DataFrame → Export CSV File : DataFrame name.to_csv("file path (file name)")

pandas.DataFrame.to_csv

```
DataFrame.to_csv(path_or_buf=None, sep=',', na_rep='', float_format=None,
columns=None, header=True, index=True, index_label=None, mode='w', encoding=None,
compression='infer', quoting=None, quotechar='"', lineterminator=None,
chunksize=None, date_format=None, doublequote=True, escapechar=None, decimal='.',
errors='strict', storage_options=None)
```

[source]

Excel File

- **Excel file : .xls(x)**

Import Excel File → DataFrame : pandas.read_excel(“file path (file name)”)

pandas.read_excel

```
pandas.read_excel(io, sheet_name=0, header=0, names=None, index_col=None,
usecols=None, squeeze=None, dtype=None, engine=None, converters=None,
true_values=None, false_values=None, skiprows=None, nrows=None, na_values=None,
keep_default_na=True, na_filter=True, verbose=False, parse_dates=False,
date_parser=None, thousands=None, decimal='.', comment=None, skipfooter=0,
convert_float=None, mangle_dupe_cols=True, storage_options=None)        [source]
```

Excel File

- **Excel file : .xls(x)**

DataFrame → Export Excel File : DataFrame name.to_excel(“file path (file name)”)

Multiple DataFrames → Export Excel File : pandas.ExcelWriter(“file path (file name)”)

pandas.DataFrame.to_excel

```
DataFrame.to_excel(excel_writer, sheet_name='Sheet1', na_rep='',
float_format=None, columns=None, header=True, index=True, index_label=None,
startrow=0, startcol=0, engine=None, merge_cells=True,
encoding=_NoDefault.no_default, inf_rep='inf', verbose=_NoDefault.no_default,
freeze_panes=None, storage_options=None)                                [source]
```

pandas.ExcelWriter

```
class pandas.ExcelWriter(path, engine=None, date_format=None,
datetime_format=None, mode='w', storage_options=None, if_sheet_exists=None,
engine_kwargs=None, **kwargs)                                           [source]
```


Thank you
Q & A