SPAI

An AI Singapore Student Chapter

ML Bootcamp

Day 2

Scan to mark attendance

Scan the QR code to mark your attendance

Attendance

SPAI

Learning Objectives

- Understanding Machine Learning, and the general machine learning workflow.
- Understanding supervised learning
- Perform data processing with Pandas and Scikit-Learn

Machine Learning Recap

What is Machine Learning:

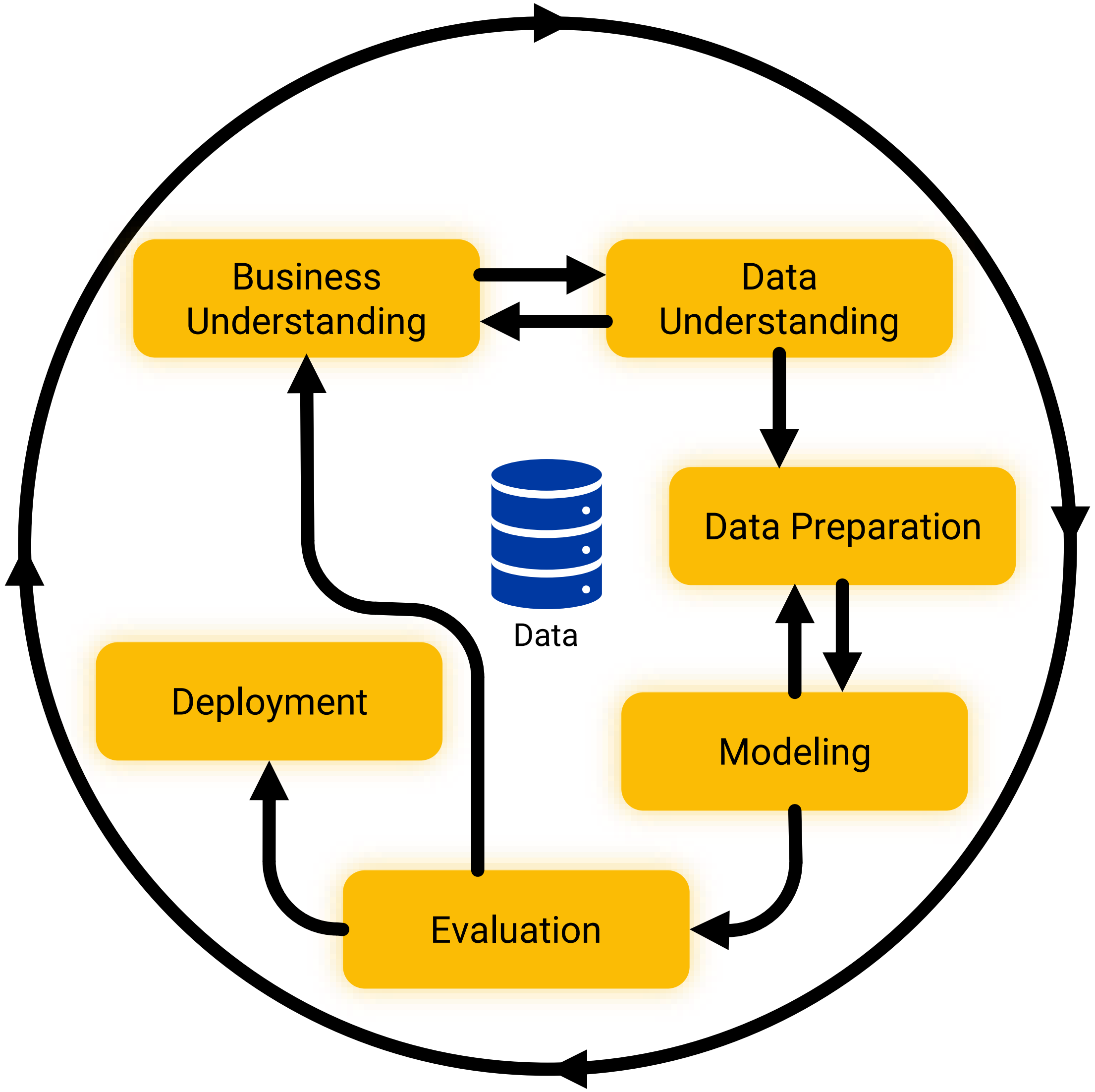
- ✅ Giving computers ability to learn from data given
- ✅ Using algorithms and statistics to analyse and draw inferences in data

Machine Learning Recap

Why Machine Learning is important

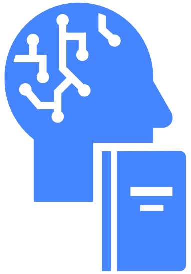
- ✅ Enables us to analyse massive quantities of data
- ✅ Allows us to better visualise those data

ML Workflow Recap



SPAI

Types of ML Algorithms



Supervised
Learning

Unsupervised
Learning

Reinforcement
Learning

SPAI

Unsupervised learning

Trained with unlabeled, uncategorized data

Output depends on coded algorithms

Two types of Unsupervised Learning

- Clustering
- Association

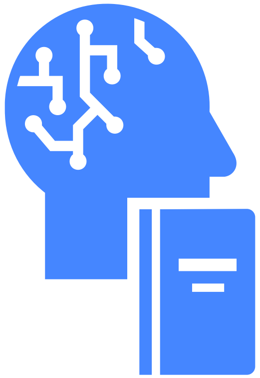
Reinforcement Learning

Trained by interacting with environment

Receives rewards by performing correctly

Receives penalties for performing incorrectly

SPAI



Supervised Learning

Trained with labelled data

Goal is to approximate mapping function so well that it can predict target/output of new input features/data

Two types of Supervised Learning

- Regression – Identifying real values
- Classification – Sorting items into categories

Supervised Learning

Regression

- To predict a quantitative outcome variable
- Goal is to build mathematical equation between outcome and input variable

Classification

- Has class labels as output like “Cat” and ”Dog”

Knowledge Check

Link the problems to the general types of machine learning algorithm required to solve them

Problem 1: You have a large inventory of products. You want to predict how many of these items will sell over the next 3 months

A. Supervised Learning: Classification
B. Unsupervised learning: Clustering
C. Supervised Learning: Regression
D. Reinforcement Learning

Knowledge Check

Link the problems to the general types of machine learning algorithm required to solve them

Problem 2: Given product orders labelled as fraudulent or non-fraudulent, predict if a new product order is fraudulent

A. Supervised Learning: Classification
B. Unsupervised learning: Clustering
C. Supervised Learning: Regression
D. Reinforcement Learning

Knowledge Check

Link the problems to the general types of machine learning algorithm required to solve them

Problem 3: Given a database of customer data, automatically discover market segments and group customers into different segments

A. Supervised Learning: Classification
B. Unsupervised learning: Clustering
C. Supervised Learning: Regression
D. Reinforcement Learning

Break and Q&A

10 Minutes

SPAI

Missing Data

- Sometimes datasets contain missing data or variables with no values

- Those variables are denoted as “NaN” in Pandas DataFrame

titanic

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

How data goes missing?

- It exists but was not collected
- It does not exist

Check missing data

Checking for missing data

```
titanic.isna().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
dtype: int64
```

Knowledge Check

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	NaN	1
1	1	85	66	29	0	26.6	0.351	31.0	0
2	8	183	64	0	0	0.0	0.672	32.0	1
3	1	89	66	23	94	28.1	0.167	21.0	0
4	0	137	40	35	168	43.1	2.288	33.0	1

>> How many features in this dataset contain missing values?

A. 1

B. 2

C. 3

D. 4

Check missing data

Missing data may have been denoted with preset value ('?' or '0')

Preset value makes it seem like there is no missing data

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	NaN	1
1	1	85	66	29	0	26.6	0.351	31.0	0
2	8	183	64	0	0	0.0	0.672	32.0	1
3	1	89	66	23	94	28.1	0.167	21.0	0
4	0	137	40	35	168	43.1	2.288	33.0	1

Handling missing data

- To ensure there are no missing values before modelling
- Two ways to handle missing values
    - Dropping Missing Values
    - Missing Value Imputation

Dropping missing values

```
df.dropna() #Drop all ROWS with missing values
df.dropna(axis = 1) #Drop all COLUMNS with missing values
df.drop(columns = [“List of Column Names”]) #Drop specific column
```

Dropping missing values

Pros

- Simple and Effective

Cons

- Prone to lose much data if too many missing values are present

```
df.dropna() #Drop all ROWS with missing values
df.dropna(axis = 1) #Drop all COLUMNS with missing values
df.drop(columns = ["List of Column Names"]) #Drop specific column
```

SPAI

Practice Time!

5 Minutes

Please attempt the practice:

Handling Missing Data using Pandas

SPAI

Times up

We will now go through the practice

SPAI

Missing Value Imputation

- Replacing missing data with substituted values
- Impute missing values with central tendency if values

```
dataset
```

```
  col1  col2  col3 col4 col5
0    2   5.0   3.0    6  NaN
1    9   NaN   9.0    0  7.0
2   19  17.0   NaN    9  NaN
```

```
dataset.mean()
```

```
  col1  col2  col3 col4 col5
0    2   5.0   3.0    6  7.0
1    9  11.0   9.0    0  7.0
2   19  17.0   6.0    9  7.0
```

Missing Value Imputation

- Other methods include ".mean()", ".median()" and ".mode()"
- Slice dataframe before calling corresponding method
- Then use ".fillna(mean/median/mode)" method to fill missing values

Missing Value Imputation

```
median_age = titanic['Age'].median()
fare_median = titanic['Fare'].median()

titanic['Age'] = titanic['Age'].fillna(median_age)
titanic['Fare'] = titanic['Fare'].fillna(fare_median)

titanic.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
dtype: int64
```

Scikit-Learn Imputation

- Simpler imputer function with basic strategies for imputing missing values
- Missing values imputed with constant value or using statistics of each column
- Types: “mean” / “median” / “most_frequent”

```
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy = 'mean')

df_impute = imputer.fit_transform(df)
```

Knowledge Check

Name	Birth Year	Death Year
Grana Merita	1908.0	1993.0
Olive White	1880.0	1960.0
Laura Francesca Saponara	2000.0	NaN
Barrie Chase	1933.0	NaN
Helen Penjam	1984.0	NaN
Beate Leiren	1977.0	NaN
Carl Jacobs	1916.0	2008.0
Artem Chigvintsev	1982.0	NaN
Raúl Filippi	1944.0	2016.0
Evan A. Stoliar	1962.0	2004.0

>> What is the best way to handle the missing values in the column “Death Year”?

A. Impute the missing value with the mean death year
B. Impute the missing value with the median death year
C. Impute with the birth year of the person + 80 years
D. Drop the rows with missing values

Practice Time!

5 Minutes

Please attempt the practice:

Basic Methods of Missing Value Imputation

Times up

We will now go through the practice

Lunch Time

1 Hour

SPAI

Feature Scaling

- A technique to make different features share similar ranges
- Common Techniques: Standardization, Min-Max Normalization

Why Scale

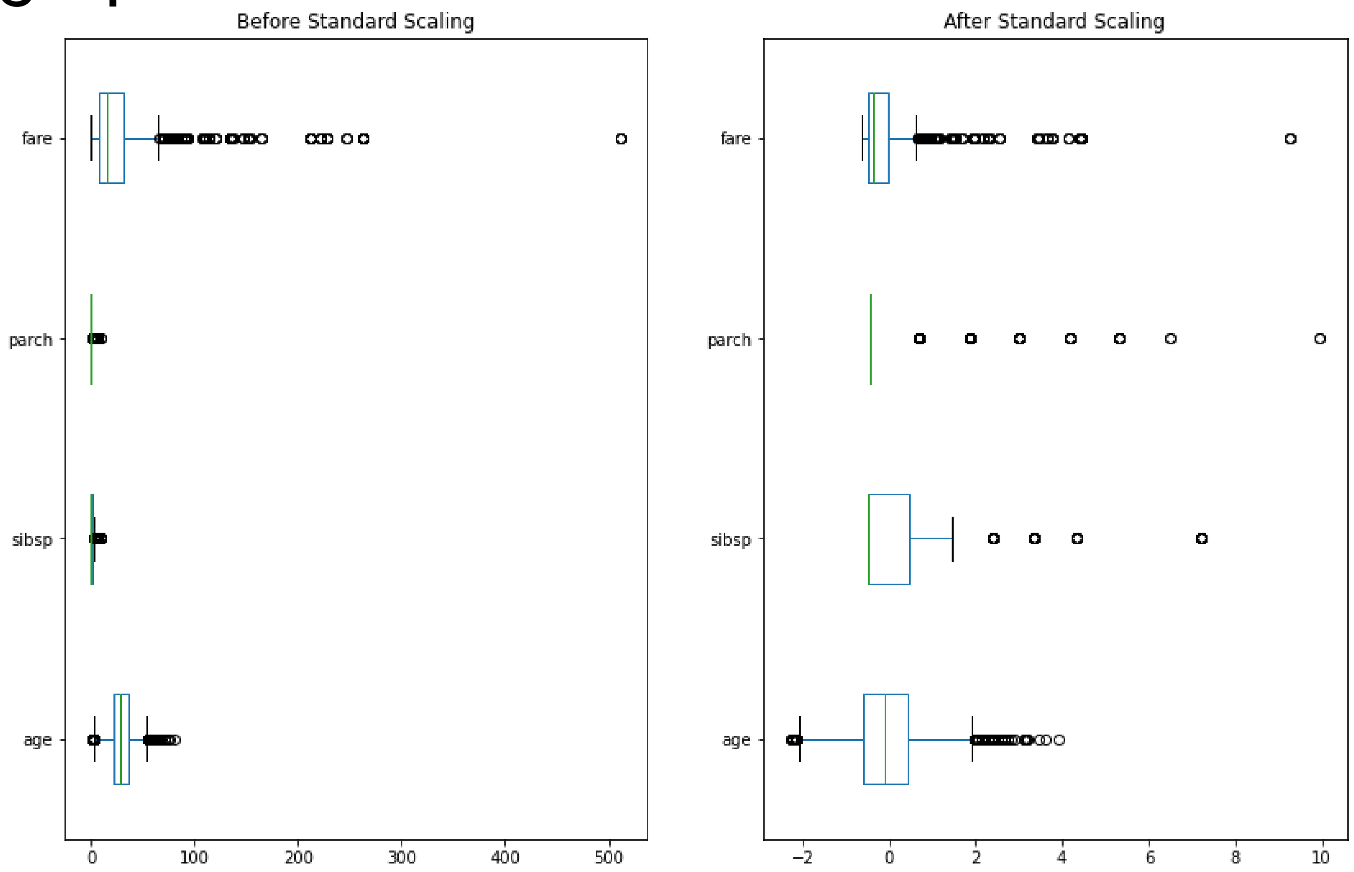
- ML algorithms are sensitive to features’ distribution.
- Algorithms tend to perform better with feature scaling
- Some of these algorithms rely on numerical optimisation methods and Distance Based algorithms.

Standardisation

- Centers data by removing mean value of each feature and scale it by dividing features by standard deviation
- Mean will be zero and standard deviation will be one

Standardisation

Makes graphs more visible



Practice Time!

5 Minutes

Please attempt the practice:

Feature Scaling

SPAI

Times up

We will now go through the practice

Break and Q&A

10 Minutes

SPAI

Outlier Recap

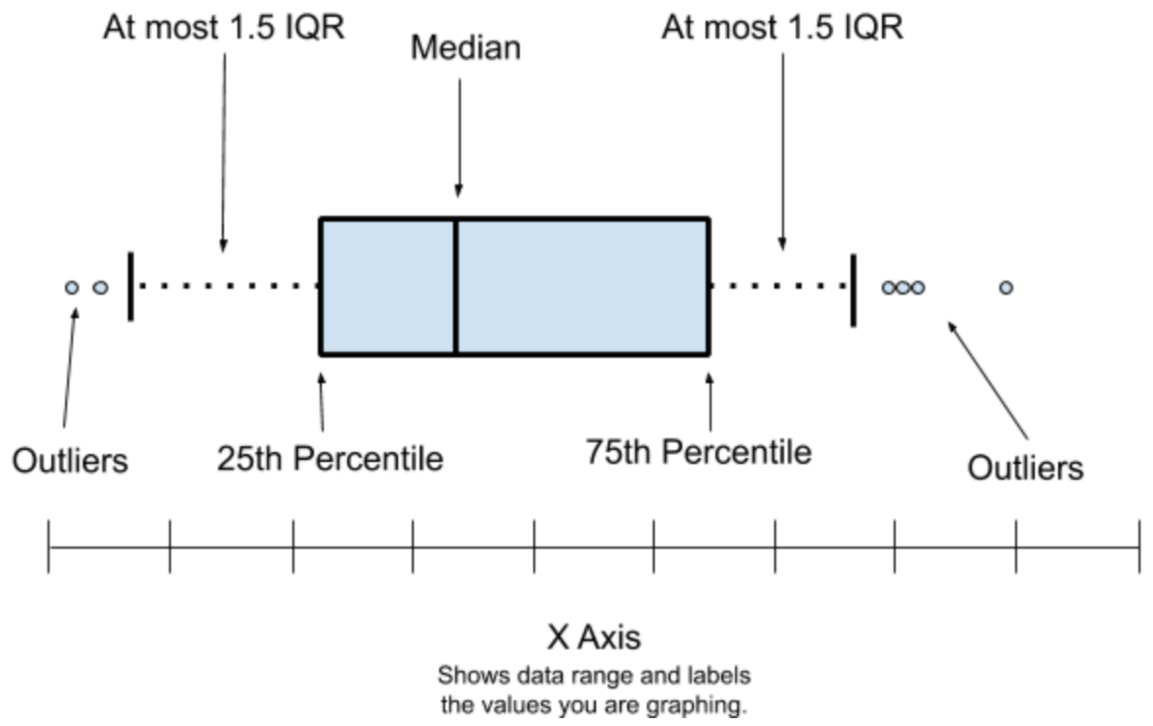
- Abnormal numerical data with extreme values.
- Machine learning algorithms might be sensitive to them
- Therefore handling outliers is important

Identifying Outliers

- Many methods; No objectively best method
- Commonly Used: Tukey Fences
- Tukey Fences: Data points 1.5 * IQR away from the upper and lower quantile are outliers.

Identifying Outliers

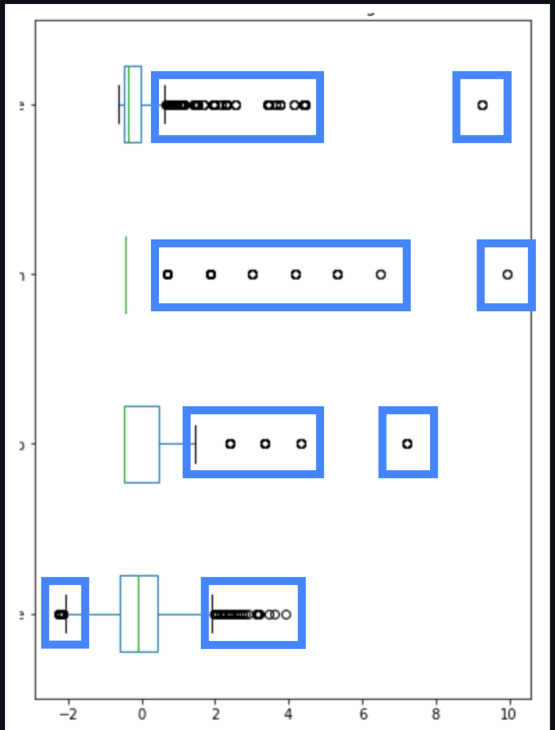
Outliers are marked using Tukey Fences in box plots



Identifying Outliers

```
import matplotlib.pyplot as plt

numerical_columns = ["Age", "SibSp", "Parch", "Fare"]
# Plotting out box plots for quantitative features
titanic[numerical_columns].plot(kind = 'box', vert = False, figsize = (12, 8))
plt.show()
```



Practice Time!

5 Minutes

Please attempt the practice:

Identifying and Removing Outliers

SPAI

Times up

We will now go through the practice

Break and Q&A

10 Minutes

Handling Categorical Features

- Converting categorical features into numerical representation
- First, determine if feature is Ordinal or Nominal
- Ordinal can be ranked/ordered, while nominal cannot

Encoding Ordinal Data

- Convert feature values to numbers where the number corresponds to the ordering of the feature
- Use OrdinalEncoder transformer from Scikit-Learn
- Fit it to a set of ordered categories and transform data to convert letter grades to numbers

```
from sklearn.preprocessing import OrdinalEncoder

enc = OrdinalEncoder()
enc.fit([[“F”, “E”, “D”, “C”, “B”, “A”]])
X[“Grade”] = enc.transform(X[“grade”])
```

Encoding Nominal Data

- No inherent ranking or order to it
- Using get_dummies method from Pandas library (Pandas creates dummy variables)
- Convert each category value into new column and assign 1 or 0 (True / Flse) to column

```
categorical_feature = ['sex', 'embarked’]
titanic_onehot = pd.get_dummies(titanic_impute, columns = categorical_feature, drop_first = True)
display(titanic_onehot)
```

Encoding Nominal Data

Pros

- Does not weight a value improperly

Cons

- Adds more columns to the data set

Encoding One Hot

- Values are converted to binary-like values
- If values below are not S or C, it implicitly must be from Q
- The original feature column should be dropped; It is left here in the image as a demonstration

	embarked	embarked_Q	embarked_S
510	S	0	1
511	Q	1	0
512	C	0	0
513	C	0	0
514	S	0	1
515	S	0	1
516	S	0	1
517	S	0	1
518	S	0	1
519	S	0	1
520	C	0	0
521	S	0	1
522	S	0	1
523	S	0	1
524	C	0	0

Knowledge Check

How should I process a dataset with missing values in a categorical feature (colour)?

A. One hot encode the data, then impute the missing value

B. Impute the missing value, then ordinally encode the data

C. Impute the missing value, the one hot encode the data

D. Ordinally encode the data, then impute the missing value

Scan to mark attendance

Scan the QR code to check out

Check Out

SPAI