NUS Fintech Society

Machine Learning Department
Training Wing

Session 2: Introduction (12/9/2020)

Content

- Github
- Data Science Project Pipeline
- Data Manipulation

Github

Git vs Github

Git	Github
Your actual face	Facebook
Command line	Website
Local version control	Share code with others

Github Basics

Repository

Fork

Branch

Clone

Pull request

Merge conflict

Merge

Master

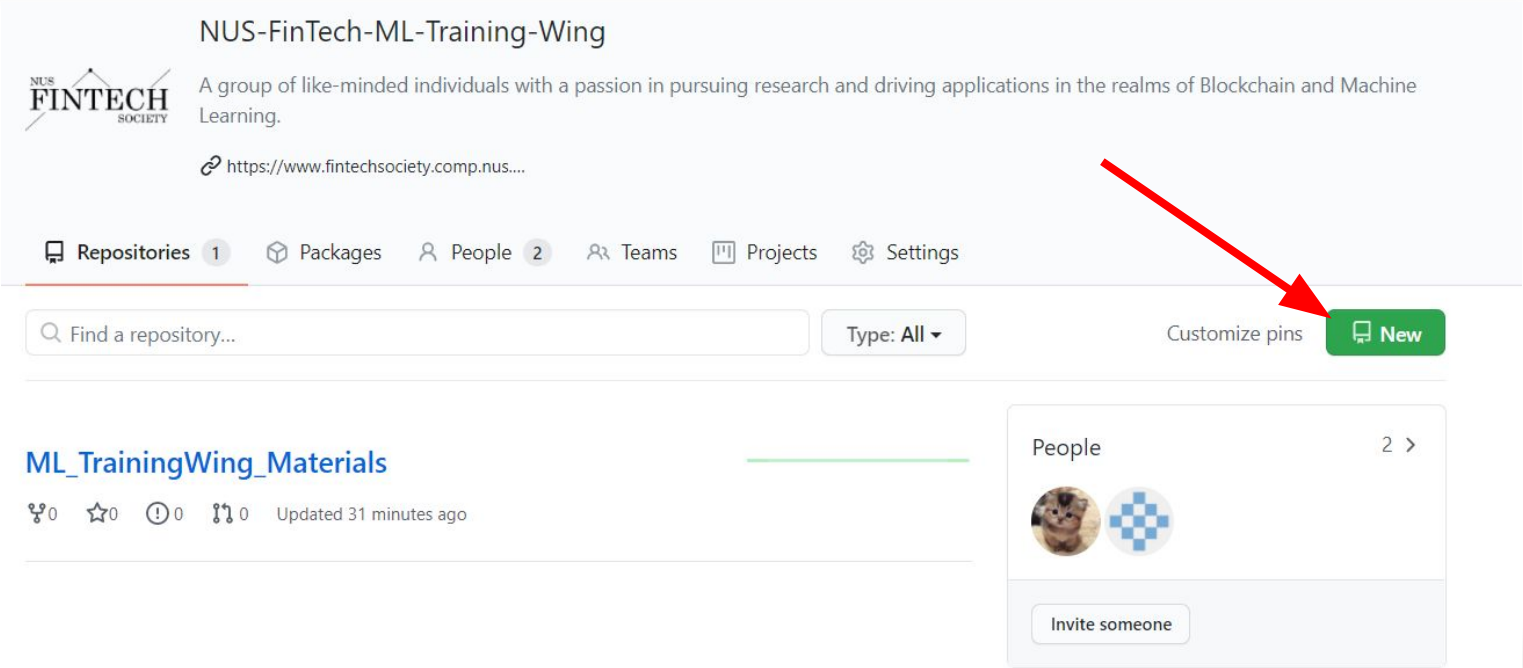
Commit

Diff

Exercise(15 mins)

- Download Github Desktop
- Create your Github Account now
- Please share your Github usernames in Slack!
- Eg. @0mission
- We will be adding you to this team now:
- https://github.com/NUS-FinTech-ML-Training-Wing
- Accept the invitation via your email.

Create a new repository



https://github.com/NUS-FinTech-ML-Training-Wing

Create a new repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
Import a repository.

Owner *

NUS-FinTech-ML-Training-Wing /

Repository name *

Zhu_Bingjie_Fintech_ML

Great repository names are short and memorable. Need inspiration? How about **sturdy-succotash**?

Description (optional)

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

- [x] Add a README file
  This is where you can write a long description for your project. Learn more.
- [ ] Add .gitignore
  Choose which files not to track from a list of templates. Learn more.
- [ ] Choose a license
  A license tells others what they can and can't do with your code. Learn more.

This will set `master` as the default branch. Change the default name in NUS-FinTech-ML-Training-Wing's settings.

Data Science Project Pipeline

Data Science Project Pipeline

1 — 2 — 3 — 4 — 5 — 6

Obtain Data

Clean Data

EDA

Feature Engineering

Modelling

Evaluation

1. Obtain Data

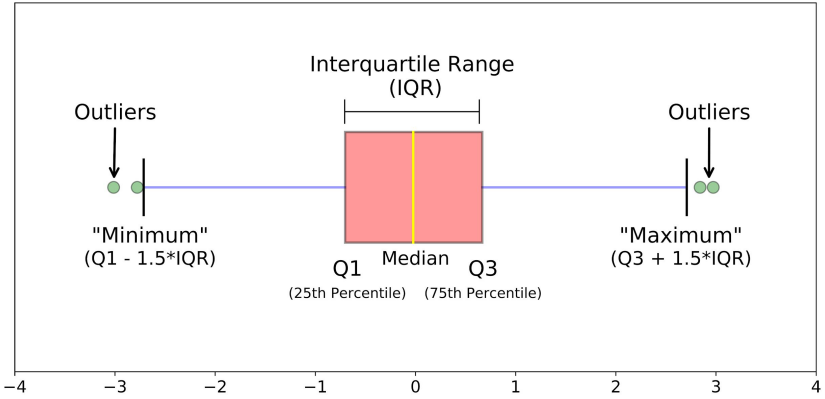
- Retrieve from Database (pyodbc)
- Web Scraping
- Excel, csv...

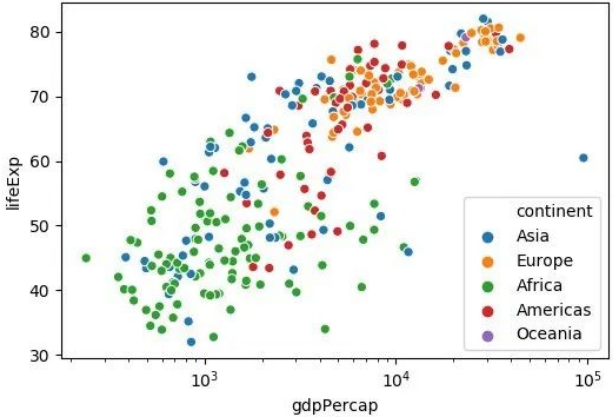
Data can be in the form of text, number, photo, audio, video...

2. Clean Data

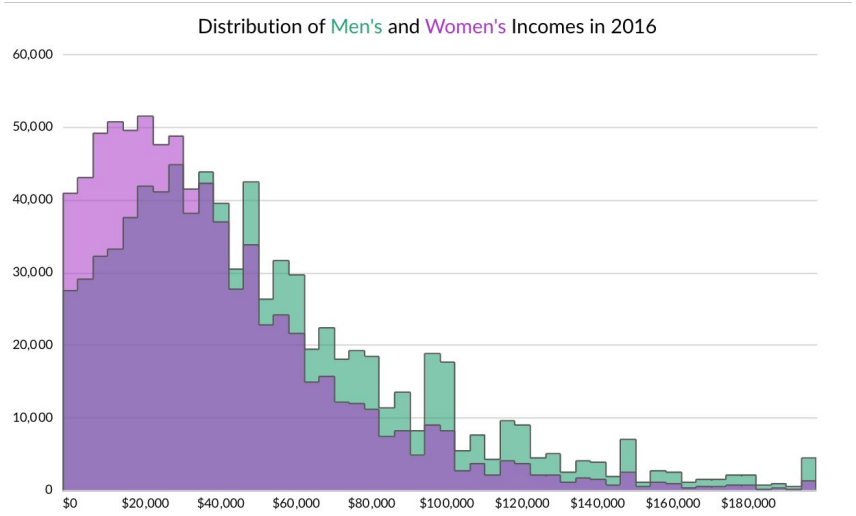
- Understand data (context is very important)
- Data quality check (validity, completeness, uniformity...)
- Deal with incomplete data (replace with mean, median, mode; delete the row)
- Dropping duplicates, dropping unnecessary columns, renaming columns, convert date format...

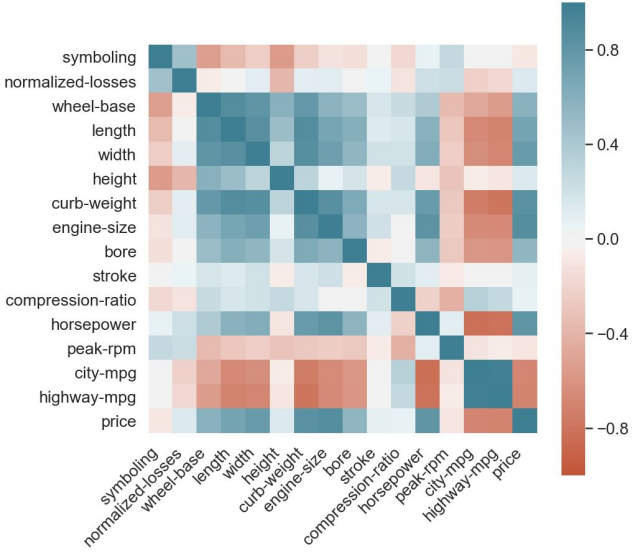
3. Exploratory Data Analysis





3. Exploratory Data Analysis

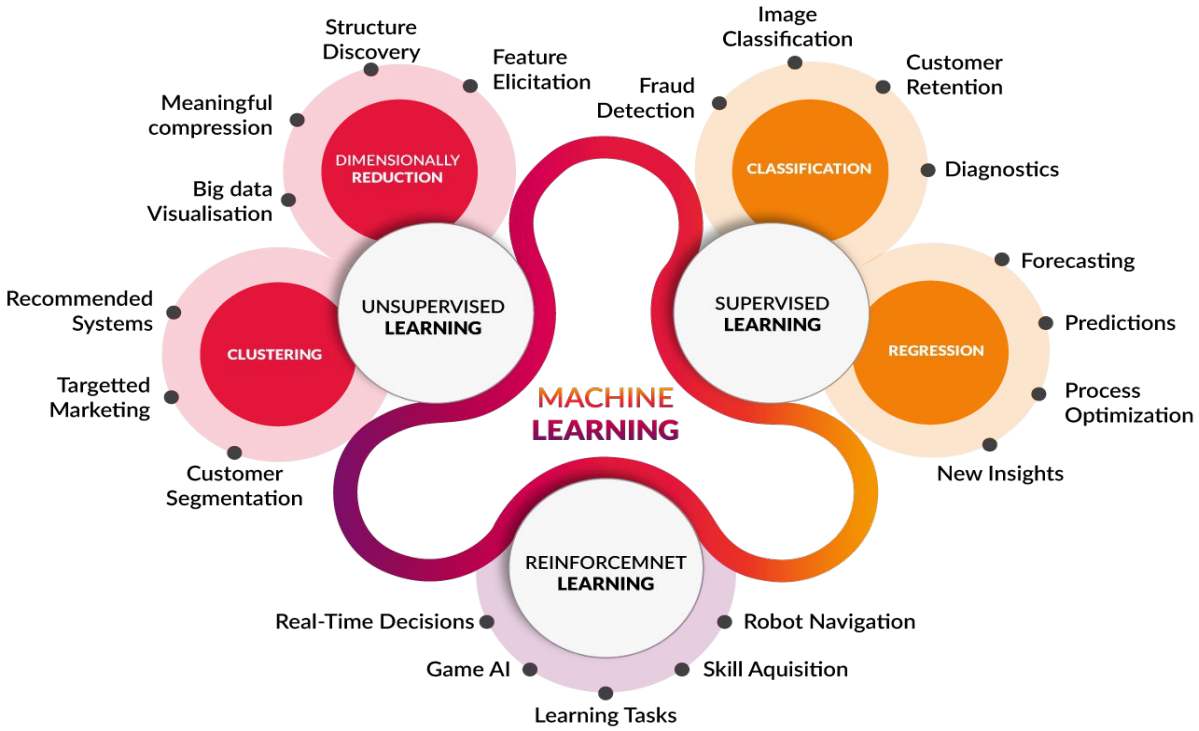




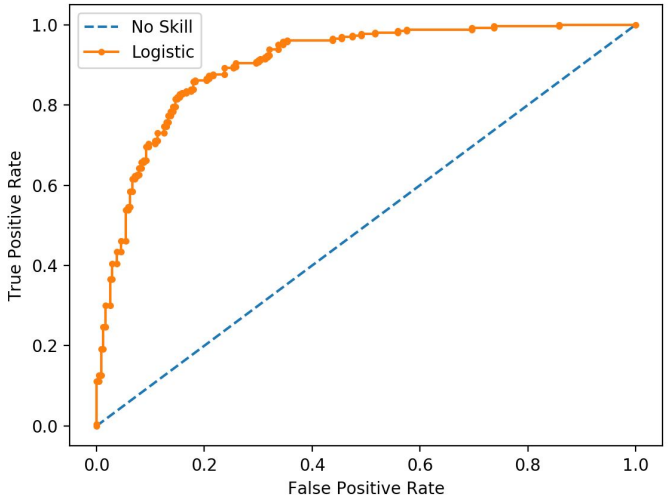
4. Feature Engineering

- Feature Selection
- Feature Transform
- Feature Extraction

5. Modelling


MACHINE LEARNING
UNSUPERVISED LEARNING
DIMENSIONALLY REDUCTION
Structure Discovery
Feature Elicitation
Meaningful compression
Big data Visualisation
CLUSTERING
Recommended Systems
Targetted Marketing
Customer Segmentation
SUPERVISED LEARNING
CLASSIFICATION
Image Classification
Customer Retention
Fraud Detection
Diagnostics
REGRESSION
Forecasting
Predictions
Process Optimization
New Insights
REINFORCEMNET LEARNING
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks

6. Evaluation



Confusion Matrix

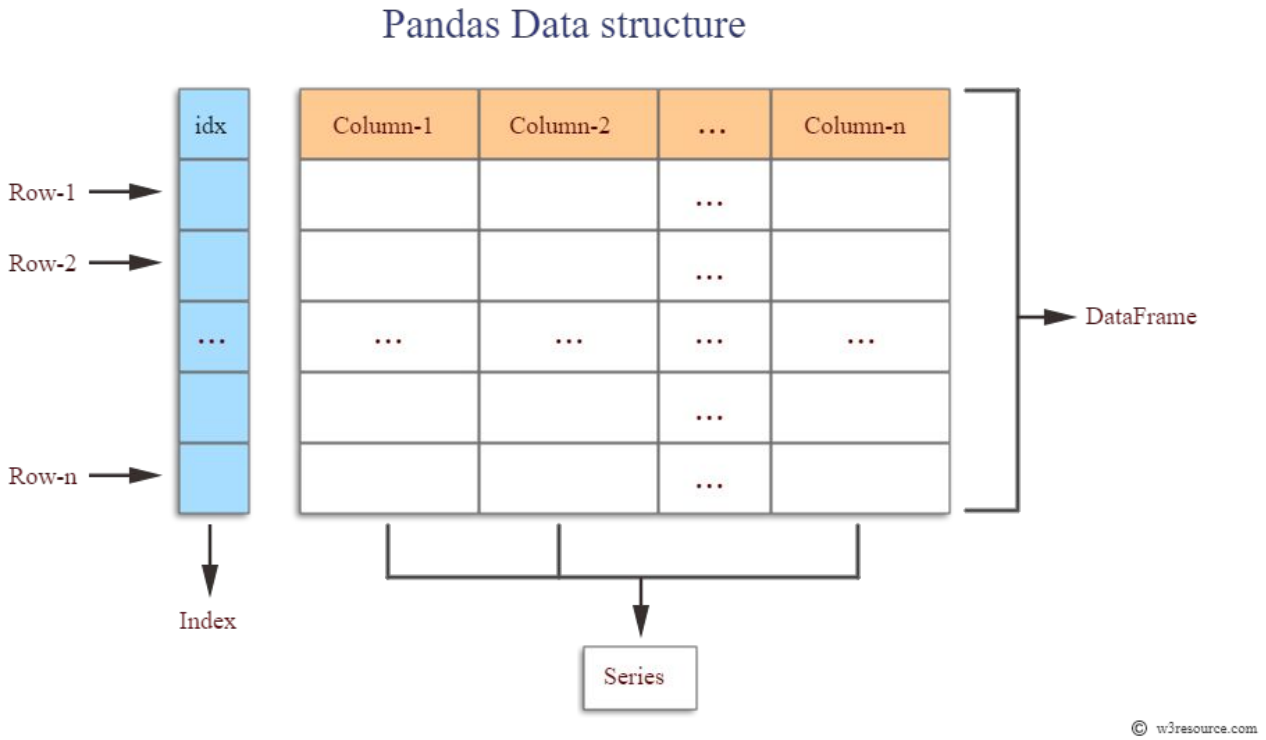
	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Accuracy, MSE, RMSE, F1 Score...

Data Manipulation

Pandas

Introduction to Pandas Data Frame



Information about the DataFrame

- Besides the df.head() function, there are other functions that retrieve information about the Data Frame easily.
- The functions are:
  1. df.sample(): Obtains 1 (default) random data from the DataFrame without replacement unless specified
  2. df.unique(): Gets an arraylist of unique values (of a column)
  3. df.nunique(): Gets the number of unique values (of a column)
  4. df.count(): Gets the count of non-NA values for each column
  5. df.min(): Get the minimum value for each column of the df
  6. df.max(): Get the maximum value for each column of the df
  7. df.sum(): Get the summation of all the values for each column

Sorting the DataFrame

- Instead of manually sorting the values, these functions exist.
- The functions are:
- df.sort_values('Adj Close'): Sorts the Data Frame by the column name specified
- df.sort_values('Adj Close', ascending=False): Sorts the Data Frame by the column name specified in the ascending order
- df.sort_values(['col_1', 'col_2'], ascending=[True, False]): Sorts the dataframe by column: 'col_1', followed by column: 'col_2'

Accessing the DataFrame

- From accessing a specific column to a specific cell, there are functions that help you to achieve it.
- To access the rows:
  1. df.iloc[0], df.iloc[0:10], df.iloc[[1,2,3]], : Locates a selection based on the position index. Unlike df.loc, df.iloc only accepts integers. For the conditions below, standard logic operations apply, such as ~ for NOT, & for AND, | for OR
  2. df.loc[df.Close >= 285]: Find all the rows based on any condition in a column
  3. df.loc[(df.Close >= 285) & (df.Open >= 285)] : Find all the rows with more than one condition
  4. df.loc[(df.Close >= 285), ['Date', 'Close']] : Select only required columns with a condition

Accessing the DataFrame

- To access the rows(Cont'):
  1. df.loc[(df.Close >= 285), ['Volume']] = 1 : Update the values of a particular column on selected rows
  2. df.loc[(df.Close >= 285), ['Adj Close', 'Volume']] = [0,1] : Update the values of multiple columns on selected rows
- To access the columns:
  1. df['Adj Close']: Access a single columns
  2. df[['Date', 'Adj Close']]: Access multiple columns
- To access a specific cell:
  1. df.loc[0, 'Adj Close']: Access a single cell

Handling empty values in the Data Frame

- There may be empty values or 'NaN'(Not a Number) values in the Dataframe.
- Let us first simulate empty values by making a cell Nan using df.loc[0, 'Adj Close'] = float("NaN")
- We will create a deep copy of the original DataFrame, to avoid making changes to it.
- To handle these empty values:
  1. df.fillna(0): Replaces all NA / NaN values with the specified value
  2. df.dropna(): Drops rows with any NA / NaN values
  3. df.isnull(): Returns a same sized object that shows if the value is a NA / NaN. If value is null, it will be shown as True else, False

Removing/Adding items

- The following functions allow you to remove/add rows/columns:
  1. df.drop([0, 1]): Drops a row by index
  2. df.drop(columns=['Open', 'Close']): Drops columns
  3. df['High-Low'] = df['High'] - df['Low']: Create new column using simple arithmetic
  4. df['new_col_name'] = df['col_name'].apply(function_name): Create new column using a function
  5. df['new_col_name'] = df['col_name'].apply(function_name, y= 1): Create new column using a function with named parameters

Exercise

- You will be split into breakout rooms of 4/5 people.
- Try to solve as many questions as you can as a group!
- Your group may be asked to present your answers after the group discussion.
- Learn from other people's approach to the question and feel free to suggest your answer as well.

Thank you

Merci beaucoup

Vielen Dank