Queen Mary
University of London

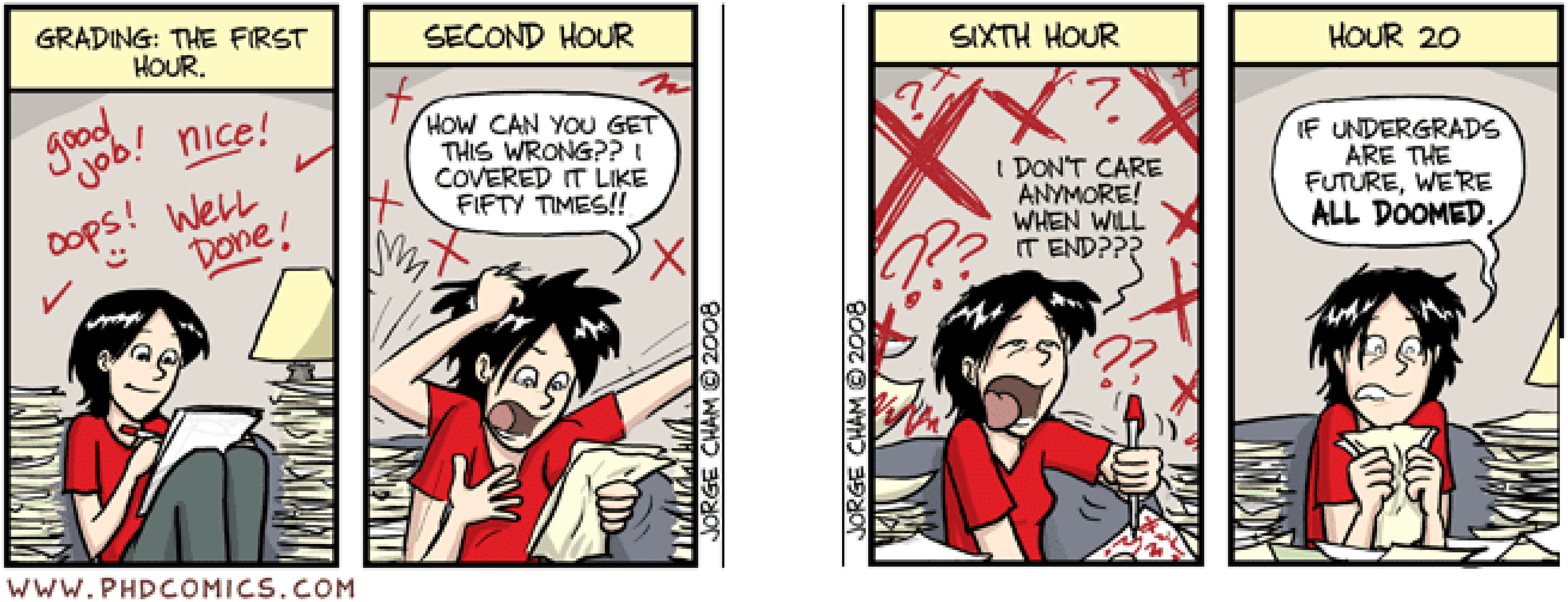
# A Tool for Detecting Similarities in Jupyter Notebooks Used as Assessment Reports

*JBEval: A Plagiarism Tool for Jupyter notebooks*

**Nikesh Bajaj**
SPCS, Queen Mary School Hainan
Queen Mary University of London
[nikesh.bajaj@qmul.ac.uk](mailto:nikesh.bajaj@qmul.ac.uk)
[https://nikeshbajaj.in](https://nikeshbajaj.in)

IEEE EDUCON 2025
IEEE Global Engineering Education Conference
London, United Kingdom || 22-25 April 2025 || Queen Mary University of London

# Pleasure and Pain of Grading…


GRADING: THE FIRST HOUR.
good job!
nice!
oops!
Well Done!
SECOND HOUR
HOW CAN YOU GET THIS WRONG?? I COVERED IT LIKE FIFTY TIMES!!
JORGE CHAM © 2008
SIXTH HOUR
I DON'T CARE ANYMORE! WHEN WILL IT END???
HOUR 20
IF UNDERGRADS ARE THE FUTURE, WE'RE ALL DOOMED.
WWW.PHDCOMICS.COM

# Overview

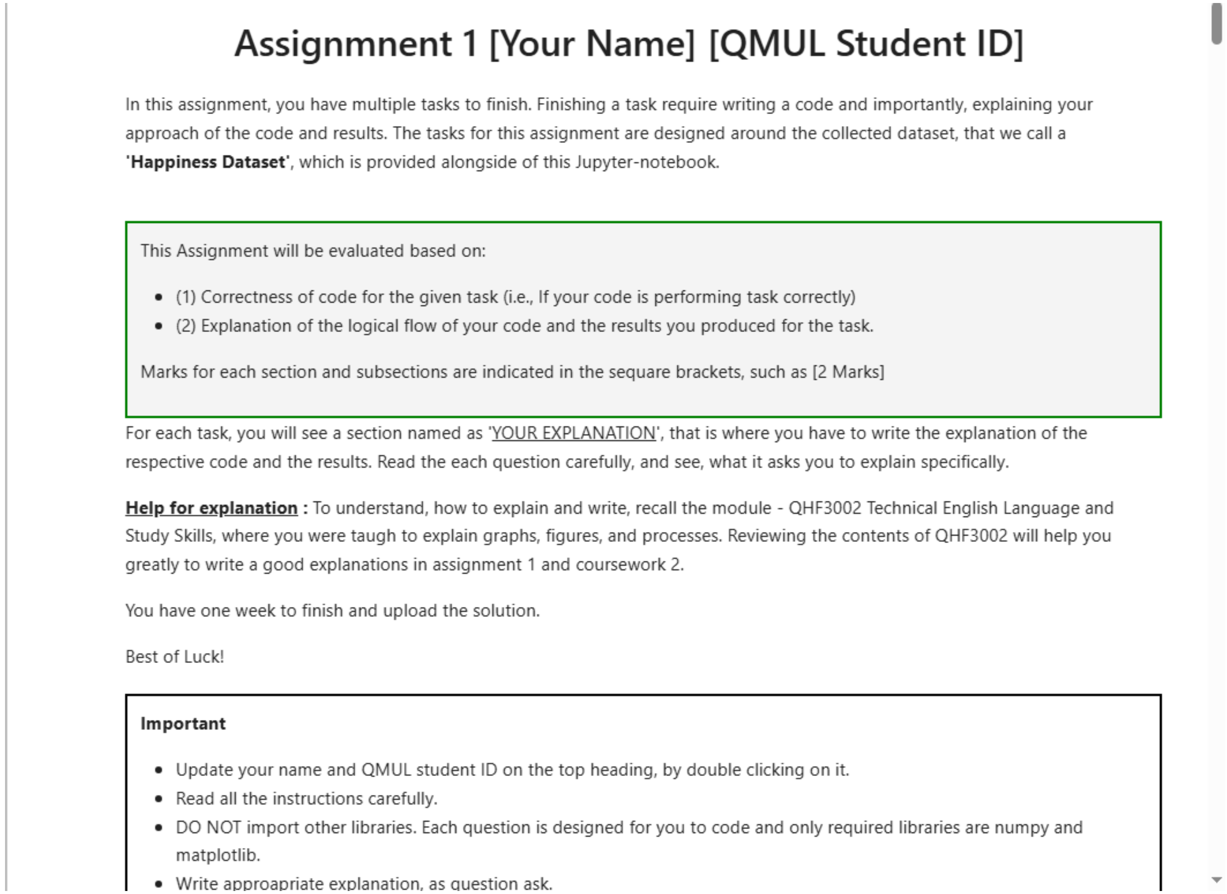
- Introduction: Jupyter Notebooks as assessments
- Plagiarism in assessments
- JBEval: Tool for Plagiarism Detection
- JBEval: Summary
- Future Work and development
- Examples: A few cases

# Introduction: Jupyter Notebooks as assessments

**Jupyter Notebooks**:

- A unique platform to assess coding-based pieces of work
- Students can write, execute code and visualise, while documenting their process through explanations, equations, and multimedia elements
- Not just programming skills
- Targets higher level of skills and contributes to Graduate Attributes of the programme
- Allowing students to explain their work in a cohesive manner- *like telling a story*.
- Educators across disciplines are increasingly adapting Jupyter Notebooks into their practice

# Jupyter Notebooks as assessments: Examples



## Assignmnent 1 [Your Name] [QMUL Student ID]

In this assignment, you have multiple tasks to finish. Finishing a task require writing a code and importantly, explaining your approach of the code and results. The tasks for this assignment are designed around the collected dataset, that we call a '**Happiness Dataset**', which is provided alongside of this Jupyter-notebook.

This Assignment will be evaluated based on:

- (1) Correctness of code for the given task (i.e., If your code is performing task correctly)
- (2) Explanation of the logical flow of your code and the results you produced for the task.

Marks for each section and subsections are indicated in the sequare brackets, such as [2 Marks]

For each task, you will see a section named as 'YOUR EXPLANATION', that is where you have to write the explanation of the respective code and the results. Read the each question carefully, and see, what it asks you to explain specifically.

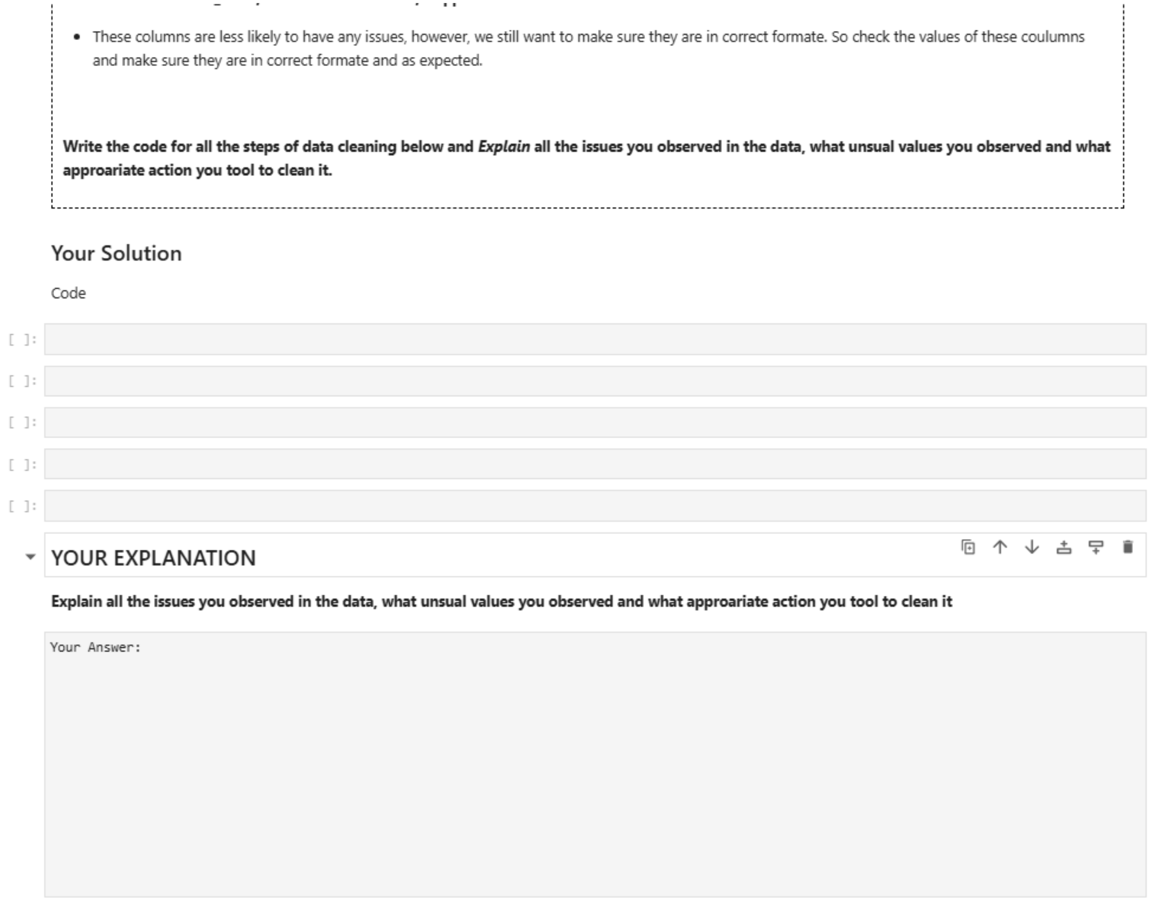
**Help for explanation :** To understand, how to explain and write, recall the module - QHF3002 Technical English Language and Study Skills, where you were taugh to explain graphs, figures, and processes. Reviewing the contents of QHF3002 will help you greatly to write a good explanations in assignment 1 and coursework 2.

You have one week to finish and upload the solution.

Best of Luck!

**Important**

- Update your name and QMUL student ID on the top heading, by double clicking on it.
- Read all the instructions carefully.
- DO NOT import other libraries. Each question is designed for you to code and only required libraries are numpy and matplotlib.
- Write approapriate explanation, as question ask.



- These columns are less likely to have any issues, however, we still want to make sure they are in correct formate. So check the values of these coulumns and make sure they are in correct formate and as expected.

**Write the code for all the steps of data cleaning below and *Explain* all the issues you observed in the data, what unsual values you observed and what approariate action you tool to clean it.**

### Your Solution

Code

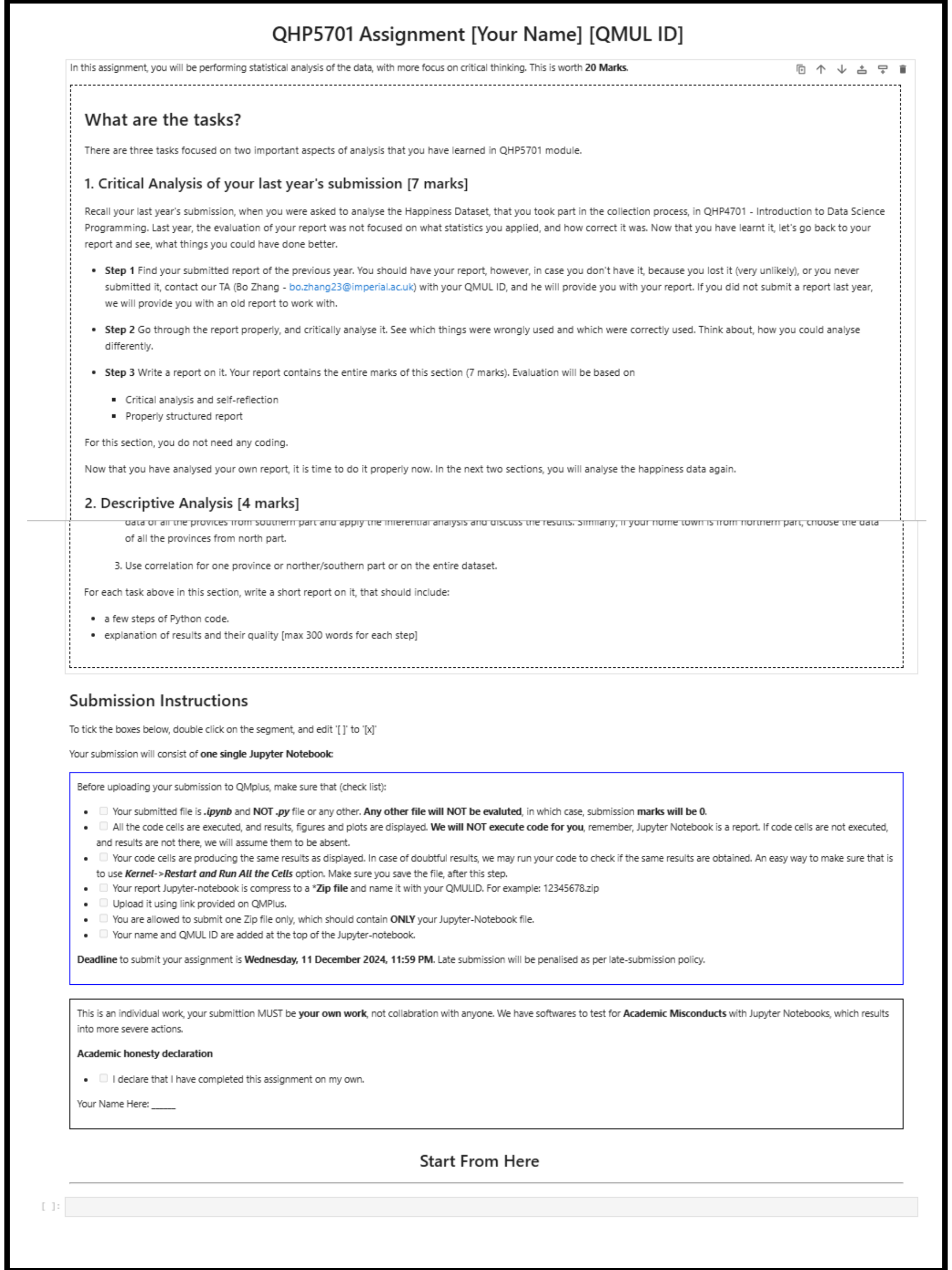
```
[ ]:
[ ]:
[ ]:
[ ]:
[ ]:
```

### YOUR EXPLANATION

**Explain all the issues you observed in the data, what unsual values you observed and what approariate action you tool to clean it**

```
Your Answer:
```

# Jupyter Notebooks as assessments: Examples



## QHP5701 Assignment [Your Name] [QMUL ID]

In this assignment, you will be performing statistical analysis of the data, with more focus on critical thinking. This is worth **20 Marks.**

### What are the tasks?

There are three tasks focused on two important aspects of analysis that you have learned in QHP5701 module.

#### 1. Critical Analysis of your last year's submission [7 marks]

Recall your last year's submission, when you were asked to analyse the Happiness Dataset, that you took part in the collection process, in QHP4701 - Introduction to Data Science Programming. Last year, the evaluation of your report was not focused on what statistics you applied, and how correct it was. Now that you have learnt it, let's go back to your report and see, what things you could have done better.

- **Step 1** Find your submitted report of the previous year. You should have your report, however, in case you don't have it, because you lost it (very unlikely), or you never submitted it, contact our TA (Bo Zhang - bo.zhang23@imperial.ac.uk) with your QMUL ID, and he will provide you with your report. If you did not submit a report last year, we will provide you with an old report to work with.
- **Step 2** Go through the report properly, and critically analyse it. See which things were wrongly used and which were correctly used. Think about, how you could analyse differently.
- **Step 3** Write a report on it. Your report contains the entire marks of this section (7 marks). Evaluation will be based on
  - Critical analysis and self-reflection
  - Properly structured report

For this section, you do not need any coding.

Now that you have analysed your own report, it is time to do it properly now. In the next two sections, you will analyse the happiness data again.

#### 2. Descriptive Analysis [4 marks]

data of all the provices from southern part and apply the inferential analysis and discuss the results. Similarly, if your home town is from northern part, choose the data of all the provinces from north part.

3. Use correlation for one province or norther/southern part or on the entire dataset.

For each task above in this section, write a short report on it, that should include:

- a few steps of Python code.
- explanation of results and their quality [max 300 words for each step]

### Submission Instructions

To tick the boxes below, double click on the segment, and edit '[ ]' to '[x]'

Your submission will consist of **one single Jupyter Notebook:**

Before uploading your submission to QMplus, make sure that (check list):

- □ Your submitted file is ***.ipynb*** and **NOT *.py*** file or any other. **Any other file will NOT be evaluted**, in which case, submission **marks will be 0**.
- □ All the code cells are executed, and results, figures and plots are displayed. **We will NOT execute code for you**, remember, Jupyter Notebook is a report. If code cells are not executed, and results are not there, we will assume them to be absent.
- □ Your code cells are producing the same results as displayed. In case of doubtful results, we may run your code to check if the same results are obtained. An easy way to make sure that is to use ***Kernel->Restart and Run All the Cells*** option. Make sure you save the file, after this step.
- □ Your report Jupyter-notebook is compress to a *__Zip file__ and name it with your QMULID. For example: 12345678.zip
- □ Upload it using link provided on QMPlus.
- □ You are allowed to submit one Zip file only, which should contain **ONLY** your Jupyter-Notebook file.
- □ Your name and QMUL ID are added at the top of the Jupyter-notebook.

**Deadline** to submit your assignment is **Wednesday, 11 December 2024, 11:59 PM.** Late submission will be penalised as per late-submission policy.

This is an individual work, your submittion MUST be **your own work**, not collabration with anyone. We have softwares to test for **Academic Misconducts** with Jupyter Notebooks, which results into more severe actions.

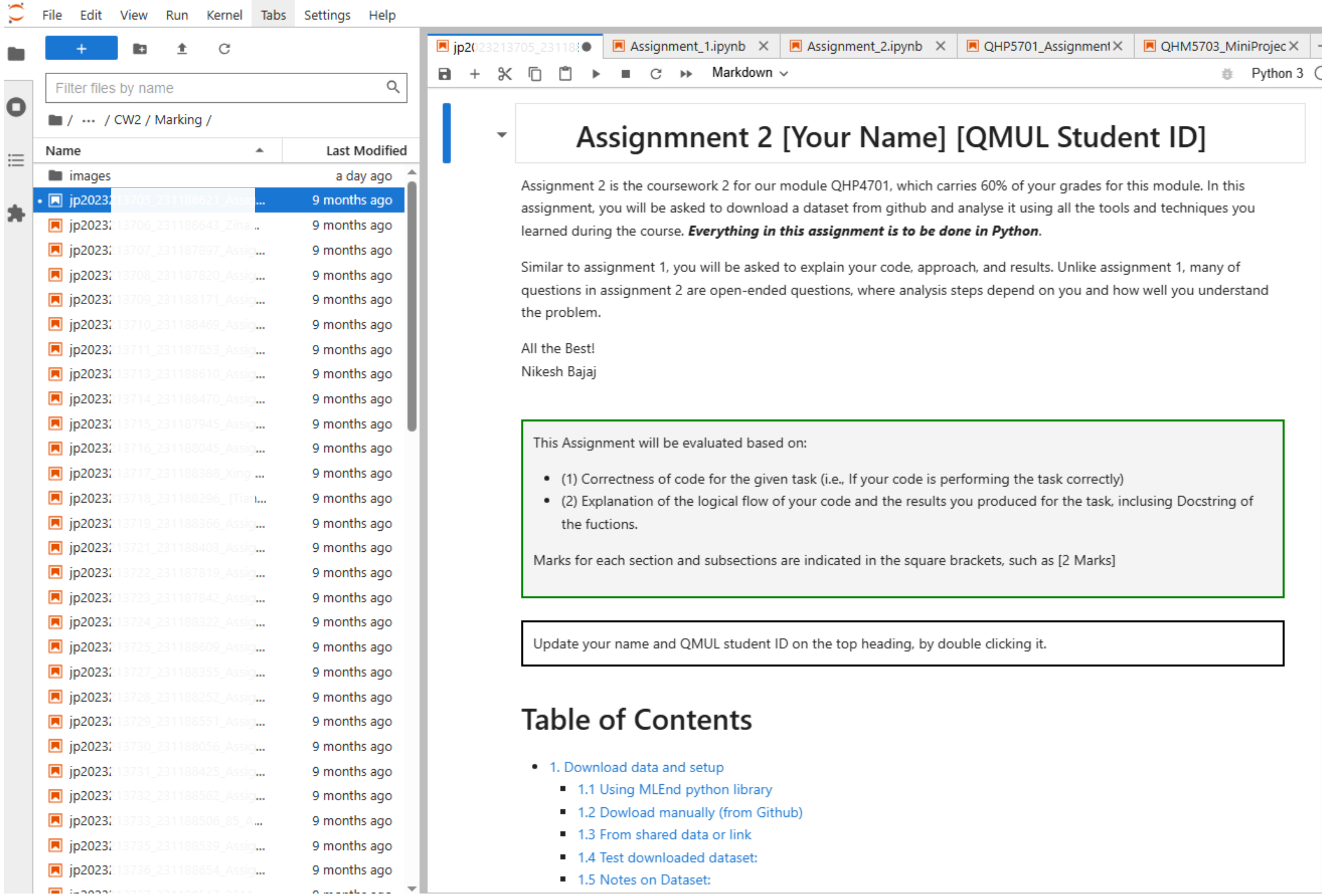
**Academic honesty declaration**

- □ I declare that I have completed this assignment on my own.

Your Name Here: ______

### Start From Here

# Grading



/ ··· / CW2 / Marking /

| Name | Last Modified |
|---|---|
| images | a day ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |
| jp2023[redacted]... | 9 months ago |

jp20[redacted]

## Assignmnent 2 [Your Name] [QMUL Student ID]

Assignment 2 is the coursework 2 for our module QHP4701, which carries 60% of your grades for this module. In this assignment, you will be asked to download a dataset from github and analyse it using all the tools and techniques you learned during the course. ***Everything in this assignment is to be done in Python***.

Similar to assignment 1, you will be asked to explain your code, approach, and results. Unlike assignment 1, many of questions in assignment 2 are open-ended questions, where analysis steps depend on you and how well you understand the problem.

All the Best!
Nikesh Bajaj

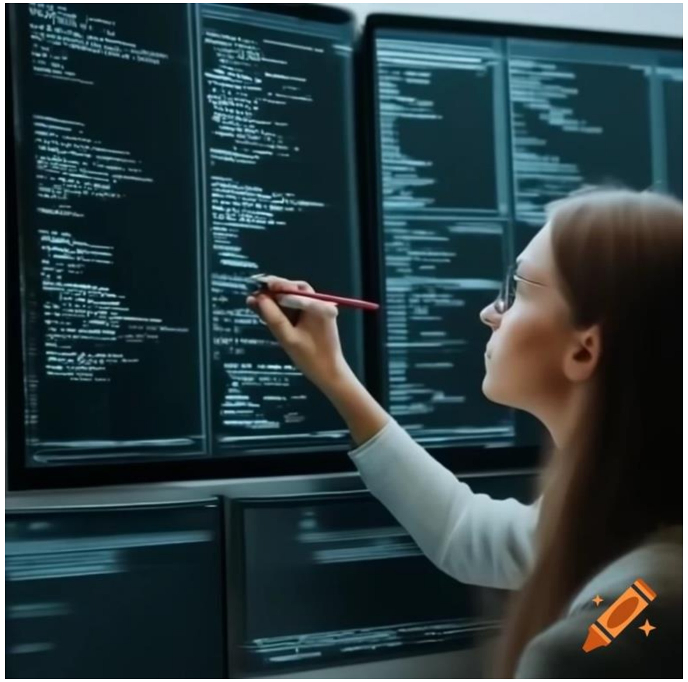
This Assignment will be evaluated based on:

- (1) Correctness of code for the given task (i.e., If your code is performing the task correctly)
- (2) Explanation of the logical flow of your code and the results you produced for the task, inclusing Docstring of the fuctions.

Marks for each section and subsections are indicated in the square brackets, such as [2 Marks]

Update your name and QMUL student ID on the top heading, by double clicking it.

## Table of Contents



# Jupyter Notebooks as assessments

**Challenges:**

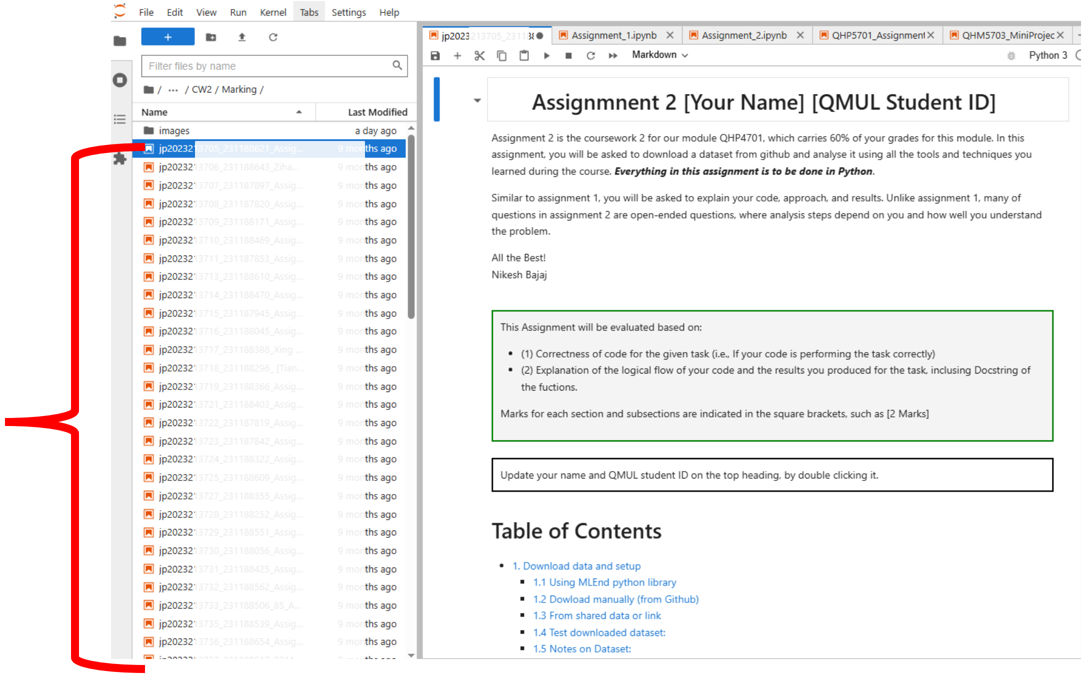
- Identifying instances of plagiarism and collusion
- Traditional plagiarism detection tools are not compatible with reports using Jupyter Notebooks
- Assessors with the ***sole responsibility*** to manually identify academic misconduct cases.
- Manual approaches could be effective in some extent. However, detecting plagiarism against previous cohort is significantly more challenging

# Plagiarism in assessments

Without a tool plagiarism detection depends on:

- **Memory**
- **Will to go through all the files**
- **Time**
- **Coffee/Tea**

…. (a manual approach)



/ … / CW2 / Marking /

| Name | Last Modified |
|---|---|
| images | a day ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |
| jp20232[redacted] | [redacted]ths ago |

## Assignmnent 2 [Your Name] [QMUL Student ID]

Assignment 2 is the coursework 2 for our module QHP4701, which carries 60% of your grades for this module. In this assignment, you will be asked to download a dataset from github and analyse it using all the tools and techniques you learned during the course. ***Everything in this assignment is to be done in Python***.

Similar to assignment 1, you will be asked to explain your code, approach, and results. Unlike assignment 1, many of questions in assignment 2 are open-ended questions, where analysis steps depend on you and how well you understand the problem.

All the Best!
Nikesh Bajaj

This Assignment will be evaluated based on:

- (1) Correctness of code for the given task (i.e., If your code is performing the task correctly)
- (2) Explanation of the logical flow of your code and the results you produced for the task, inclusing Docstring of the fuctions.

Marks for each section and subsections are indicated in the square brackets, such as [2 Marks]

Update your name and QMUL student ID on the top heading, by double clicking it.

### Table of Contents

- 1. Download data and setup
  - 1.1 Using MLEnd python library
  - 1.2 Dowload manually (from Github)
  - 1.3 From shared data or link
  - 1.4 Test downloaded dataset:
  - 1.5 Notes on Dataset:

***Difficult to account for previous years cohorts***

***Resits submissions***

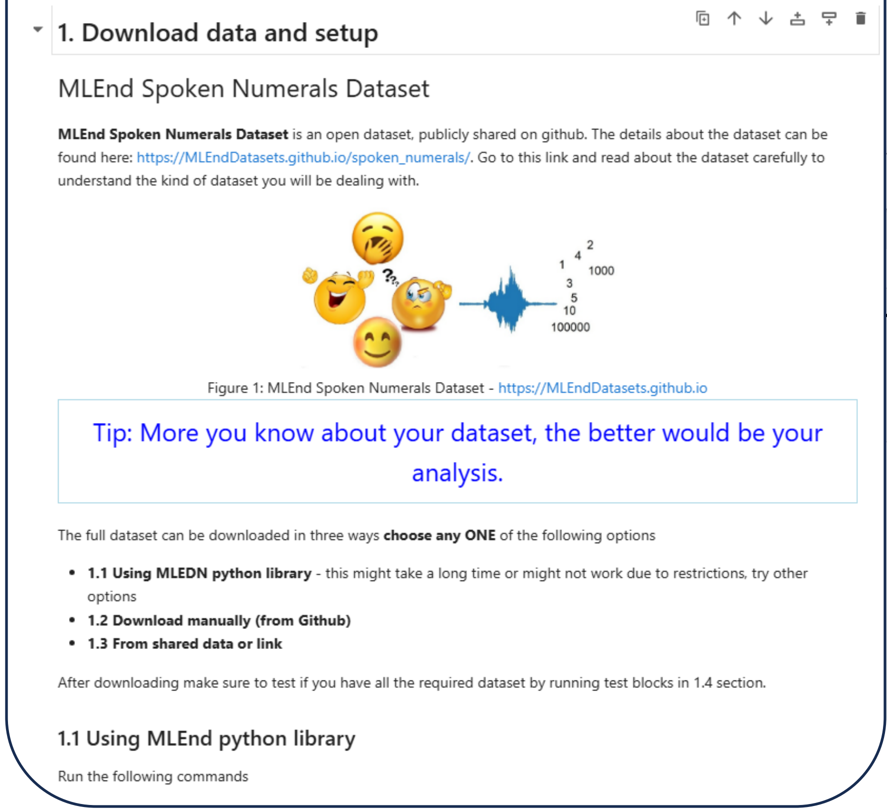
# JBEval: Plagiarism detection in Jupyter Notebooks

**JBEval:**

- A solution to identifying ***potential*** plagiarism cases
- Easy to visualise and trace back to reports to validate
- Allows to compare with previous cohort or resit

# Jupyter-Notebook

- HTML Components: Markdown $C_m$



## 1. Download data and setup

### MLEnd Spoken Numerals Dataset

**MLEnd Spoken Numerals Dataset** is an open dataset, publicly shared on github. The details about the dataset can be found here: https://MLEndDatasets.github.io/spoken_numerals/. Go to this link and read about the dataset carefully to understand the kind of dataset you will be dealing with.

Figure 1: MLEnd Spoken Numerals Dataset - https://MLEndDatasets.github.io

Tip: More you know about your dataset, the better would be your analysis.

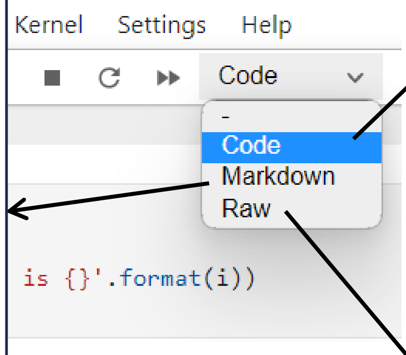
The full dataset can be downloaded in three ways **choose any ONE** of the following options

- **1.1 Using MLEDN python library** - this might take a long time or might not work due to restrictions, try other options
- **1.2 Download manually (from Github)**
- **1.3 From shared data or link**

After downloading make sure to test if you have all the required dataset by running test blocks in 1.4 section.

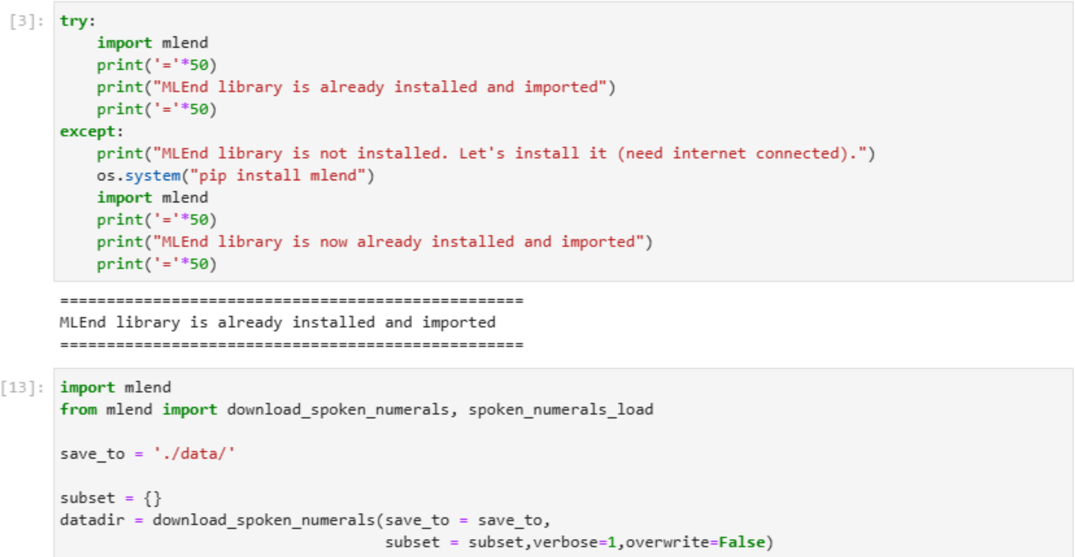
### 1.1 Using MLEnd python library

Run the following commands



```
is {}'.format(i))
```

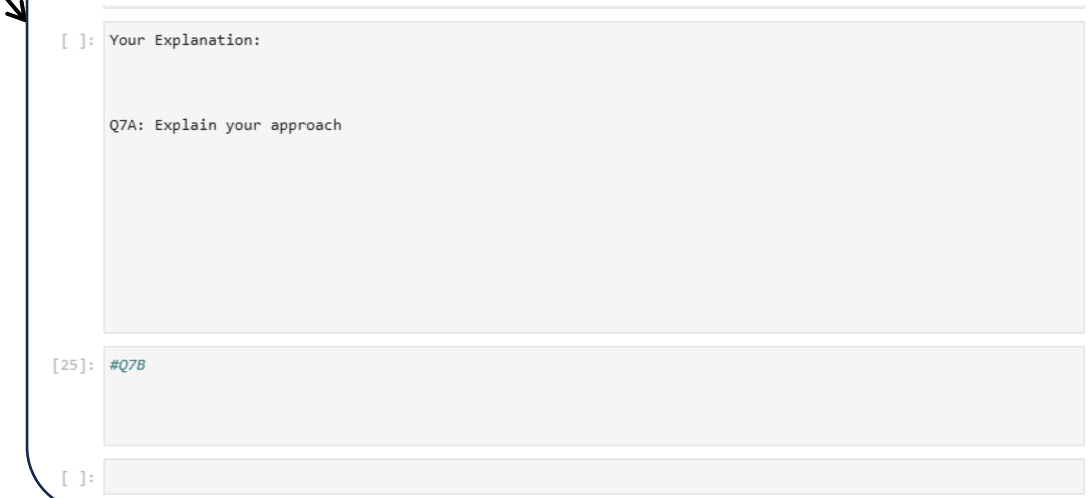
- Code Components $C_c$



```
[3]: try:
         import mlend
         print('='*50)
         print("MLEnd library is already installed and imported")
         print('='*50)
     except:
         print("MLEnd library is not installed. Let's install it (need internet connected).")
         os.system("pip install mlend")
         import mlend
         print('='*50)
         print("MLEnd library is now already installed and imported")
         print('='*50)

     ==================================================
     MLEnd library is already installed and imported
     ==================================================
```

```
[13]: import mlend
      from mlend import download_spoken_numerals, spoken_numerals_load

      save_to = './data/'

      subset = {}
      datadir = download_spoken_numerals(save_to = save_to,
                                         subset = subset,verbose=1,overwrite=False)
```

- Plain Text Components: Raw $C_p$



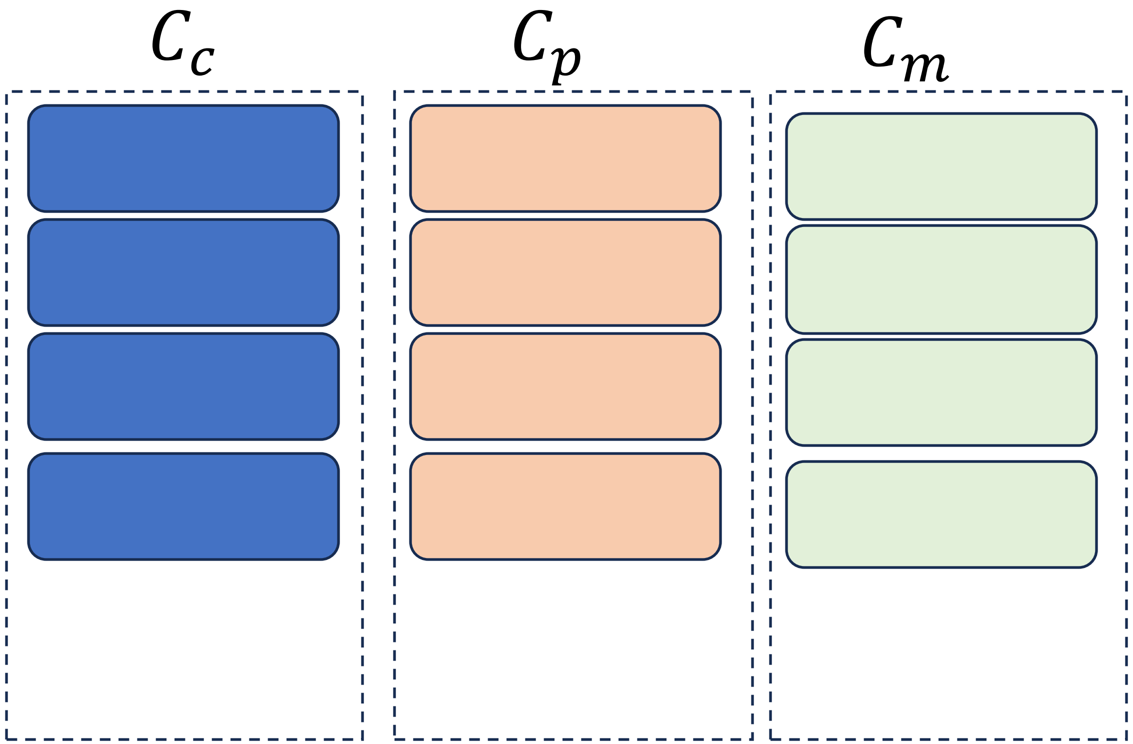
```
[ ]: Your Explanation:


     Q7A: Explain your approach
```

```
[25]: #Q7B
```

```
[ ]:
```

# Processing Jupyter-Notebooks

- **Code and Plain-Text Extraction**
  - Extract different components
  - Arrange sequentially as one component
- **Preprocessing – cleaning**
  - For code, removing comments and  spaces
  - For text, removing spaces splitting into lines
- **Similarity detection and score**
  - Computing similarity between each pair of lines



# Jaro–Winkler similarity

Jaro–Winkler similarity ***S*** is based on edit distance between two strings

$$S \in [0,1]$$

**Example**:

$c_1$ = *for i in range(10):*
$c_2$ = *for k in range(10):*
$c_3$ = *for k in LIST 10:*
$c_4$ = *for k in LIST A:*

$S_{c1,c2}$ = 0.9678
$S_{c1,c3}$ = 0.8270
$S_{c1,c4}$ = 0.6417

# Jaro–Winkler similarity with threshold $T$

We use threshold ***T***, as to consider is two strings are a match (e.g. T=0.9 )

- A threshold allows us to:
  - Detect the similarity *if code is changed with trivially* (paraphrased)

    - Change the name of variables, a few values, etc (*will see examples*)

$c_1$ = *for i in range(10):*
$c_2$ = *for k in range(10):*
$c_3$ = *for k in LIST 10:*
$c_4$ = *for k in LIST A:*

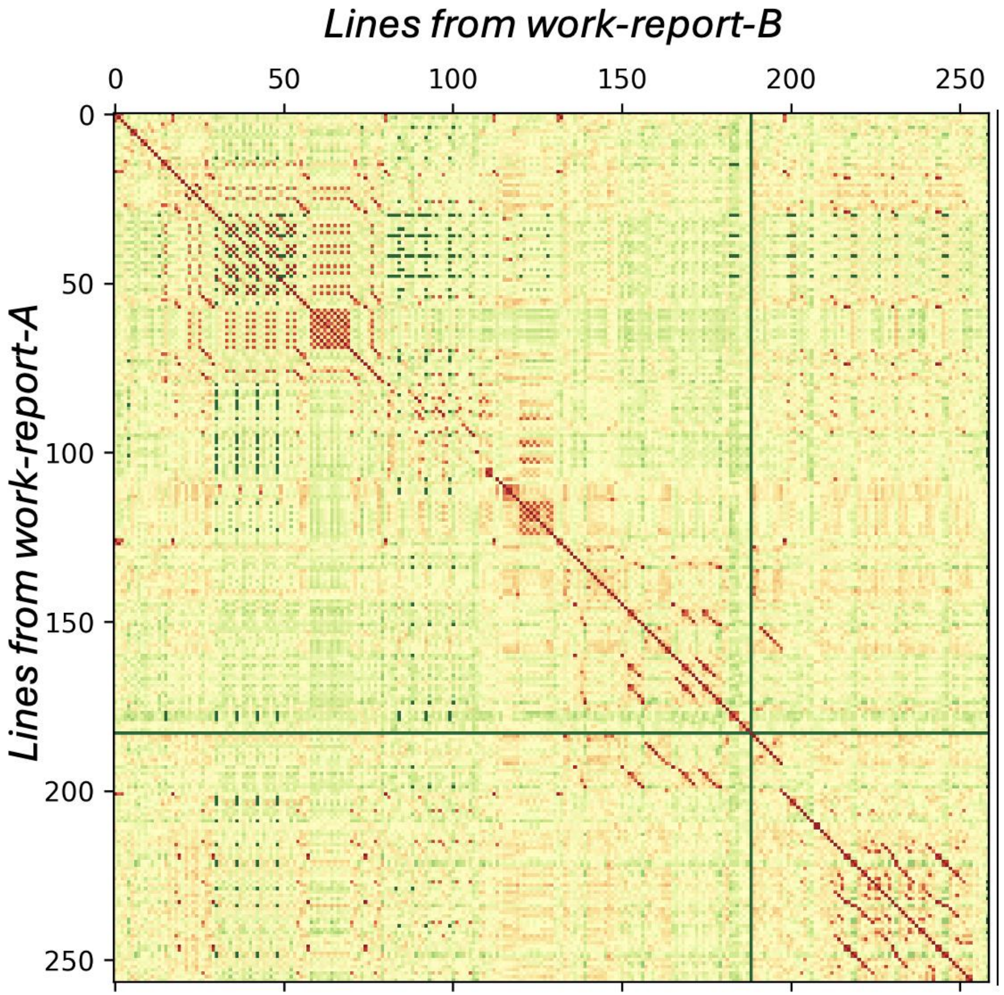
  - Detect the similarity *if text is paraphrased*, which is a usual trick to avoid plagiarism in reports (*will see examples*)

$S_{c1,c2}$ = 0.9678
$S_{c1,c3}$ = 0.8270
$S_{c1,c4}$ = 0.6417

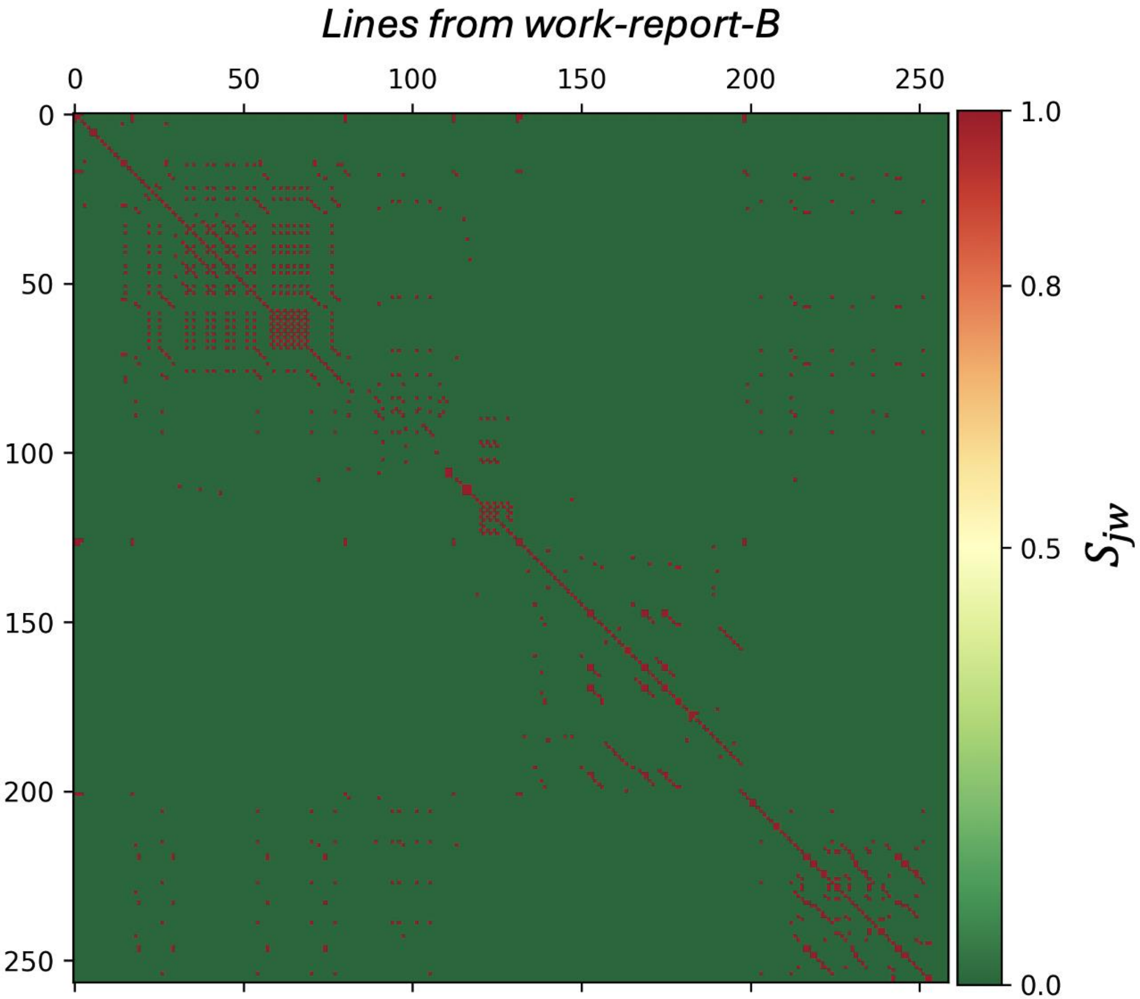
  - Adopt tool for assessments in *wide range of modules*

*Threshold **T**, allows to control the aggressiveness of the similarity detection algorithm*

# Jaro–Winkler similarity: Example

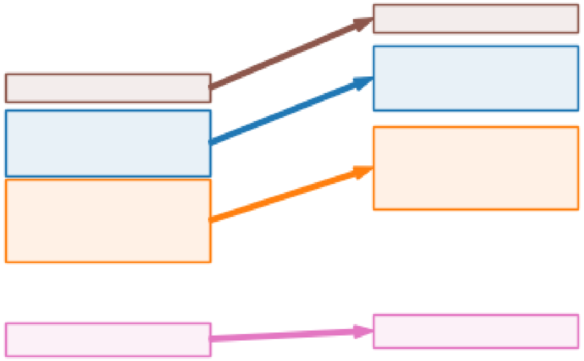


Similarity score $S_{jw}$ of each line



Line match ($S_{jw} > 0.8$)

# Block Matching

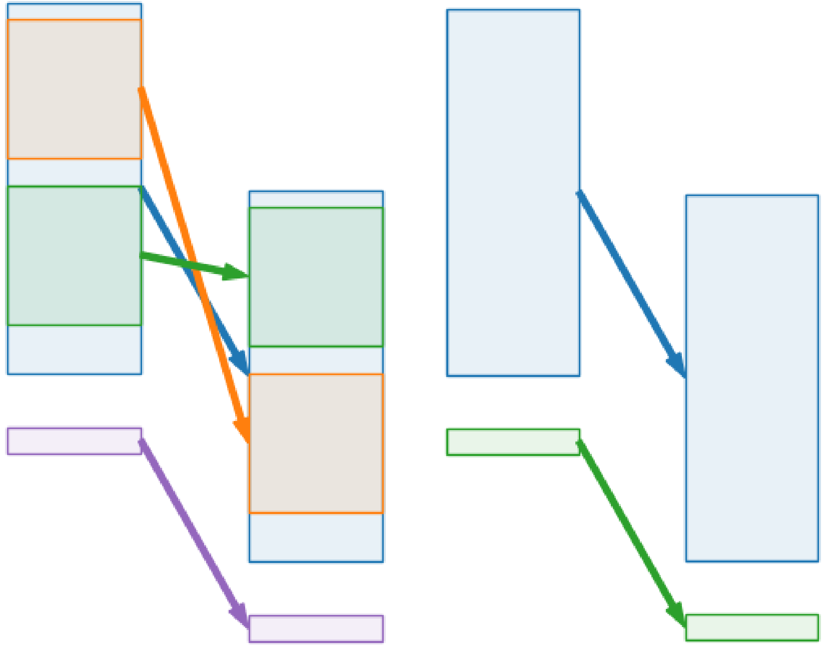


- When programming frequent tasks are encoded in similar way

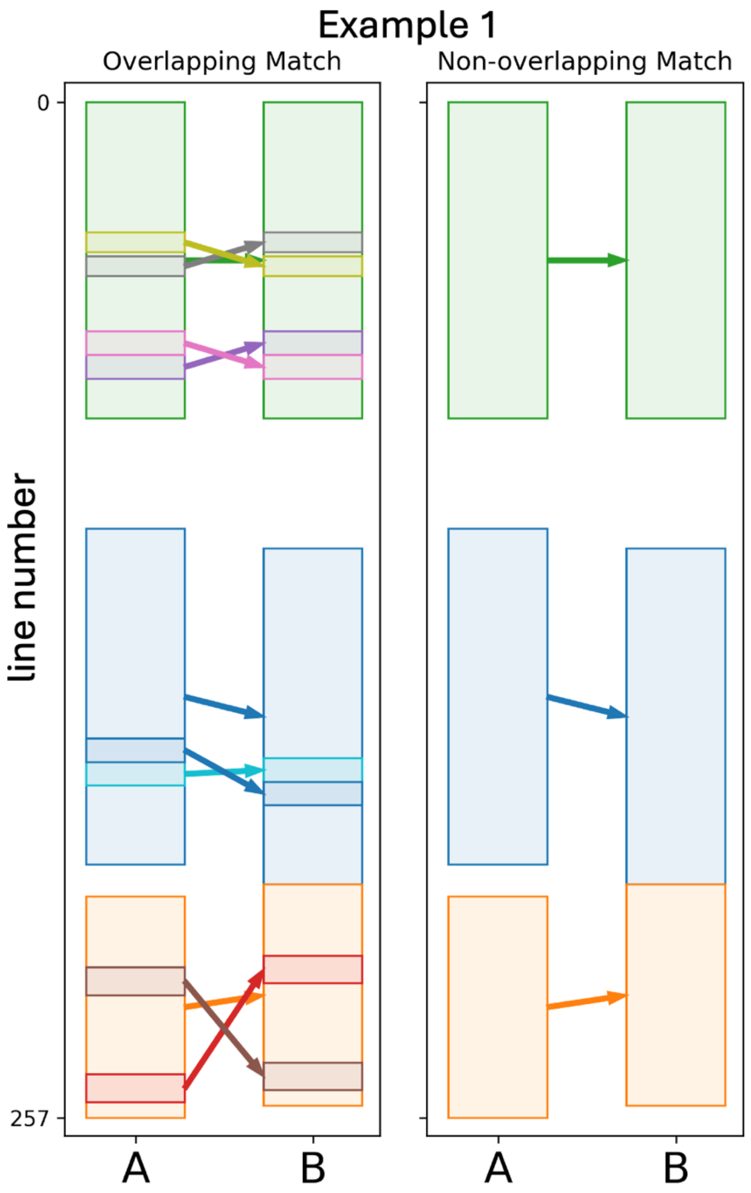
  - plotting figures, reading files and importing libraries

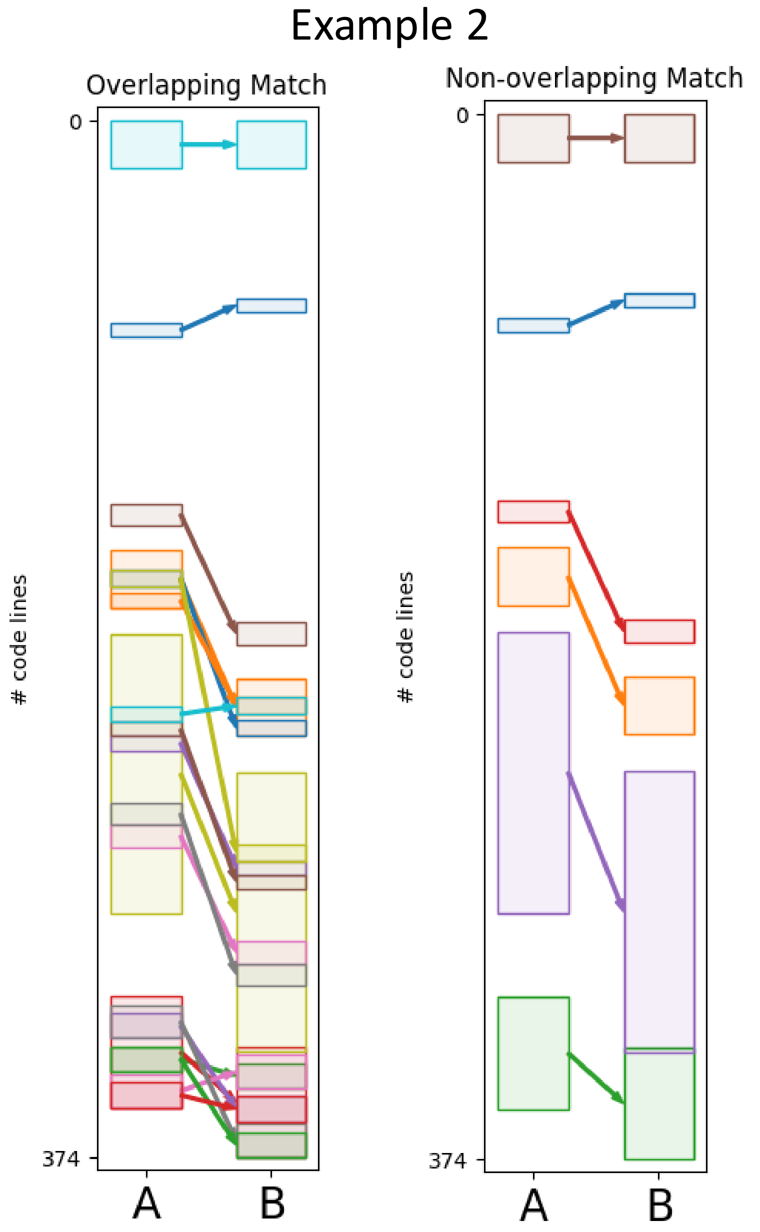
- Block-matching:
  - To avoid many instances of single line matching
  - A block of lines in report A, similar to a block of lines in report B.
  - Minimum length of block $\boldsymbol{L_m}$

- Non-overlapping Blocks:
  - Removing blocks matching multiple times



# Block Matching: Examples





# Overall Similarity Score

- Overall Similarity score of notebooks A and B

$$\text{स}_{A\leftarrow B} = \frac{\sum_j |b_j|}{|C_c|} = \frac{N_{A\leftarrow B}}{N_A}$$

where the Devanagari symbol स is pronounced as ‘*sə*’*

Similarity score:

- $\in [\mathbf{0}, \mathbf{1}]$
- *Asymmetric:* $\text{स}_{A \leftarrow B} \neq \text{स}_{B \leftarrow A}$

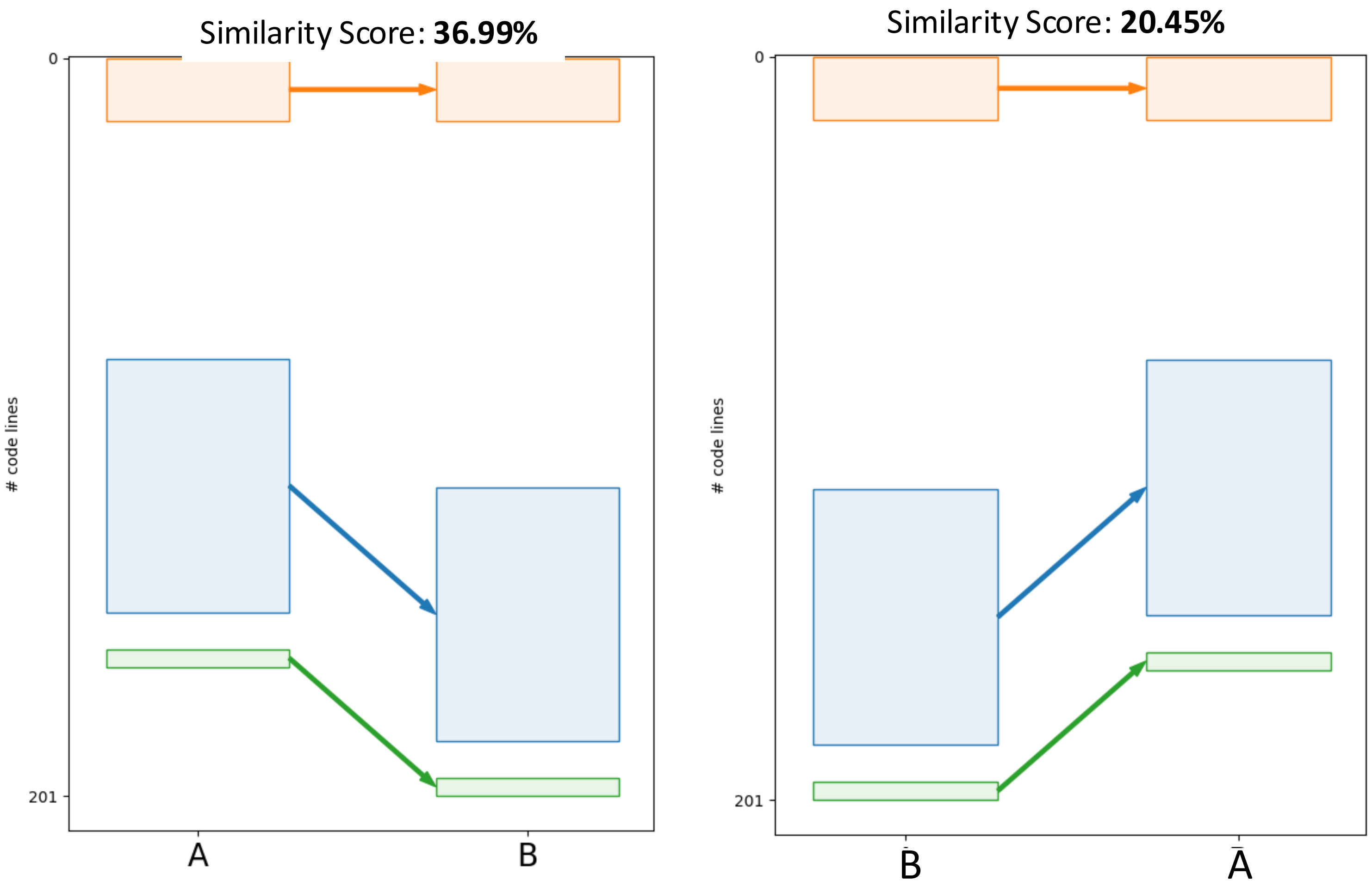
***Is similarity score enough?***

**in spirit of being inclusive*

# Overall Similarity Score and Visualisation

- Overall Similarity score of notebooks A and B

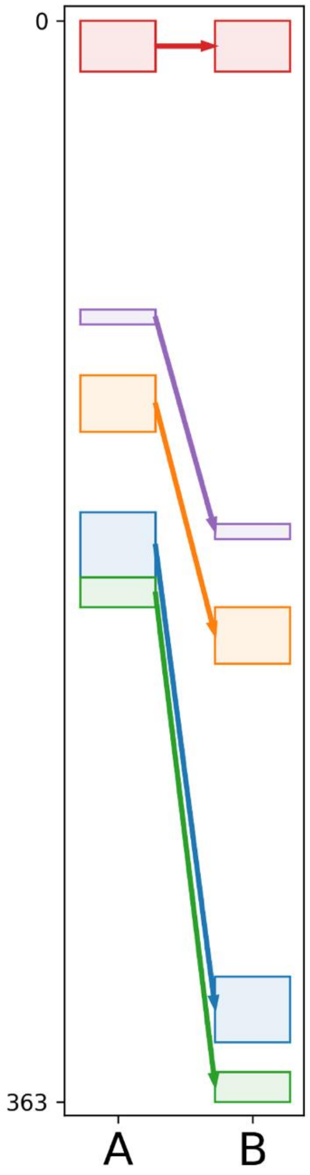
Example 1



Similarity Score: **36.99%**

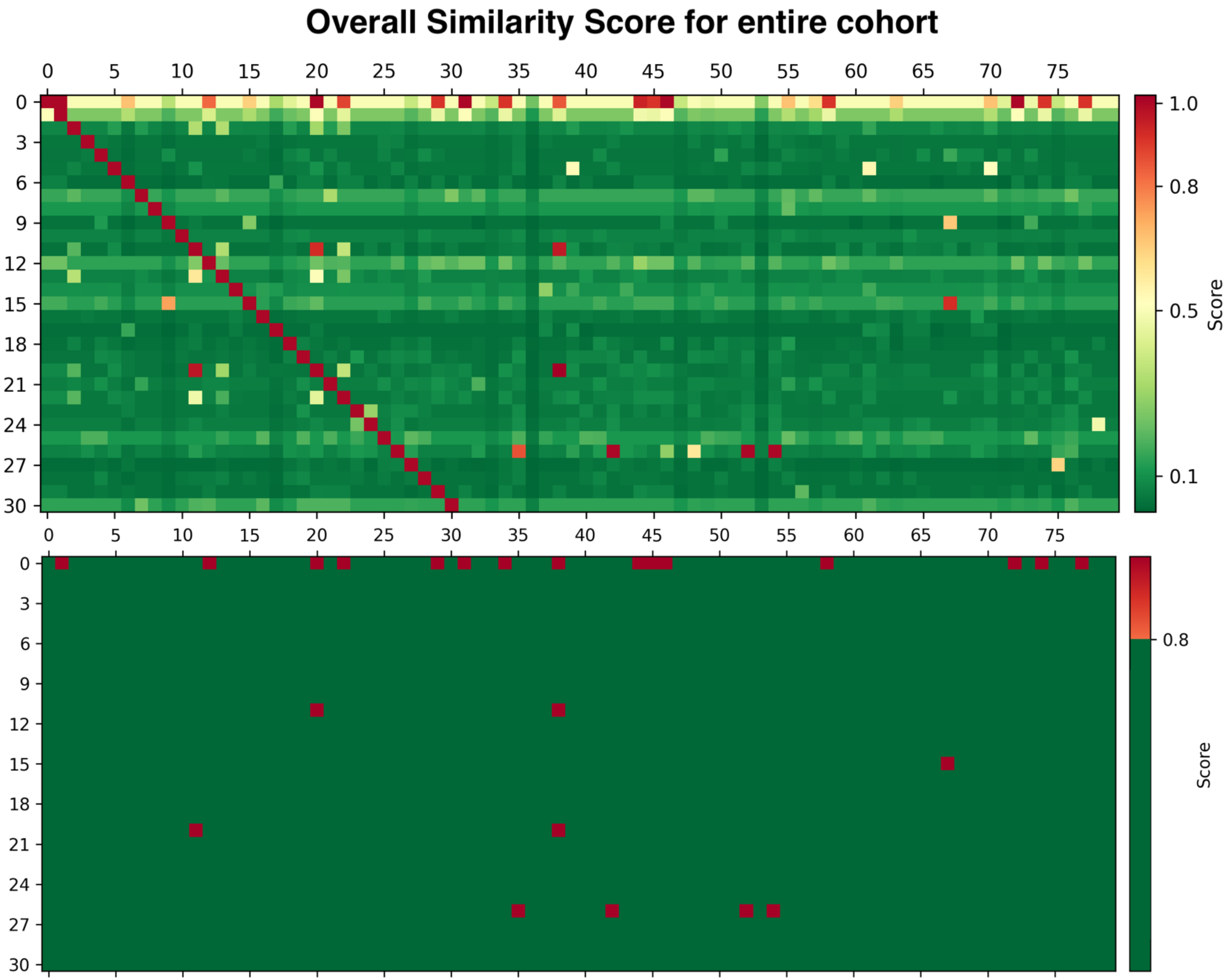
Similarity Score: **20.45%**

Example 2



# Processing entire cohort

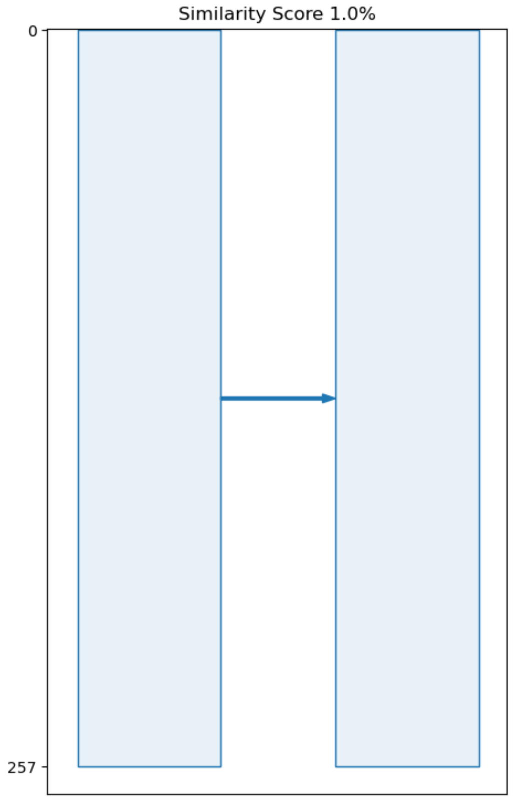
- Processing entire cohort and flagging the cases
- Comparing with previous/first-sit submissions

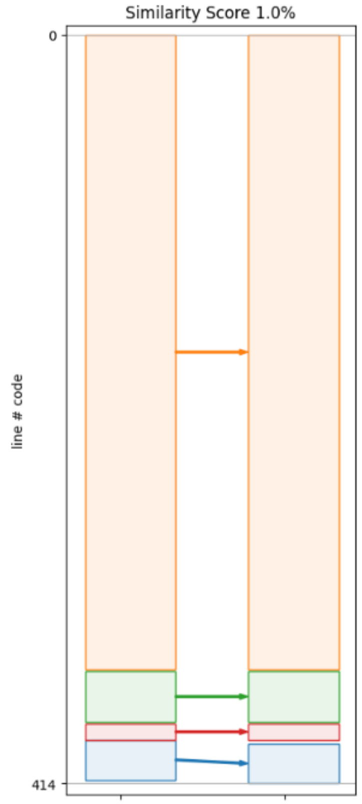


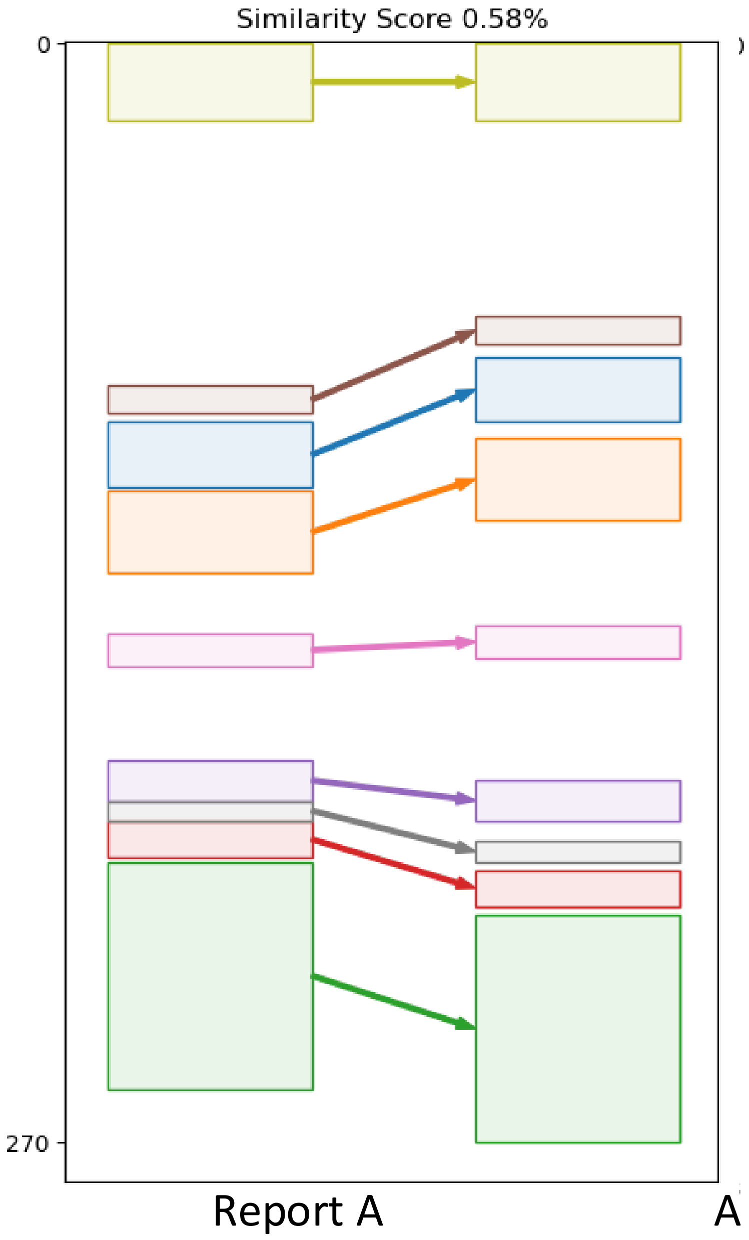
# Grouping and Visualisation

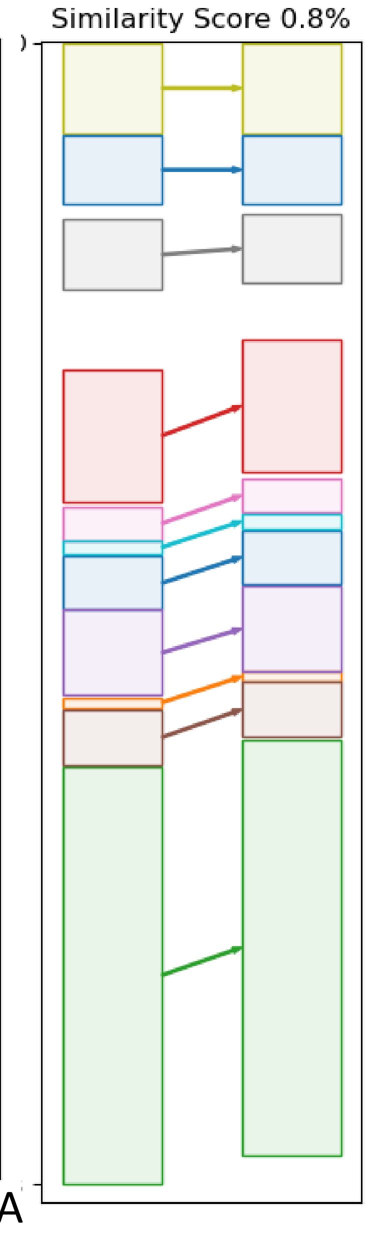
JBEval

- forms groups in case of collusion (A, B, C)
- Produces visualisation for a pair of reports for details





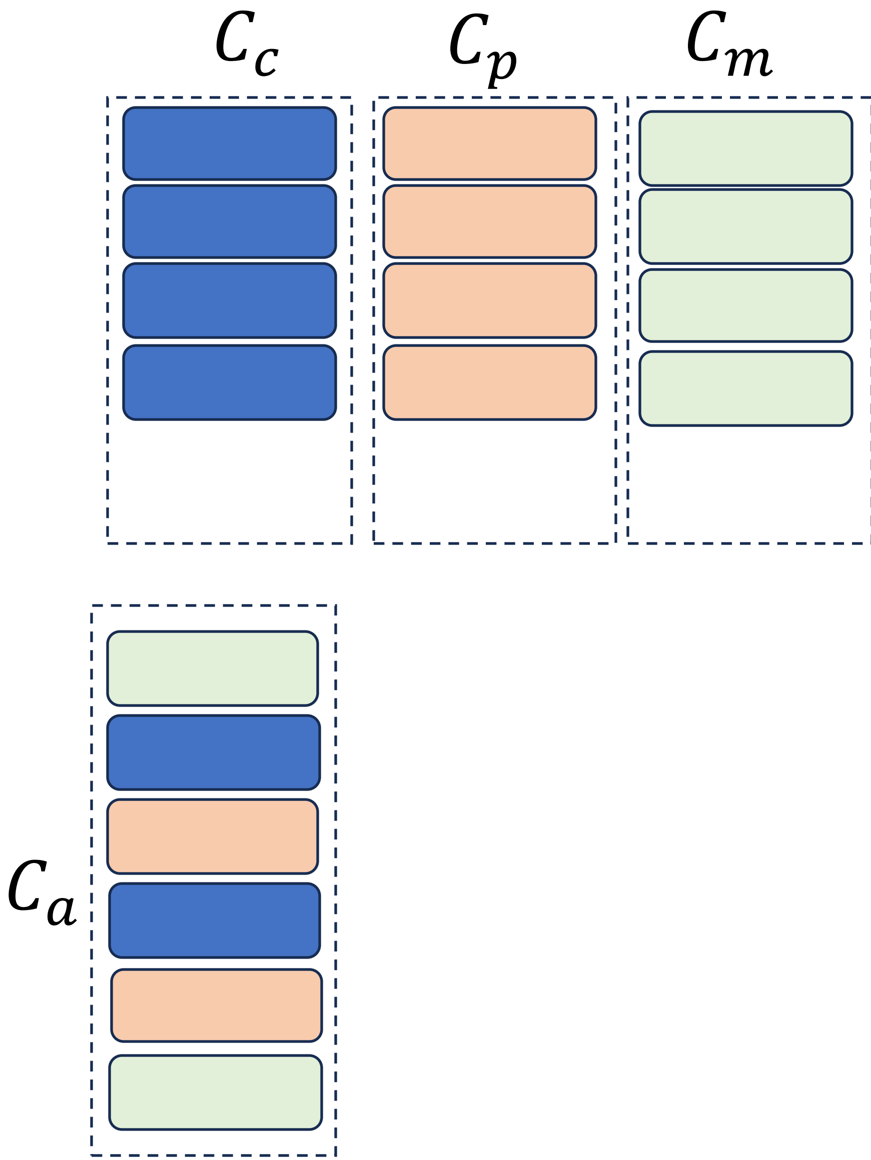




# Extension to text and markdown

- Similar to code component $C_p$, appropriate cleaning, processing and matching is applied to plain text and/or markdown text.

- The similarity score of individual components or combined can be computed.



# JBEVal: Summary

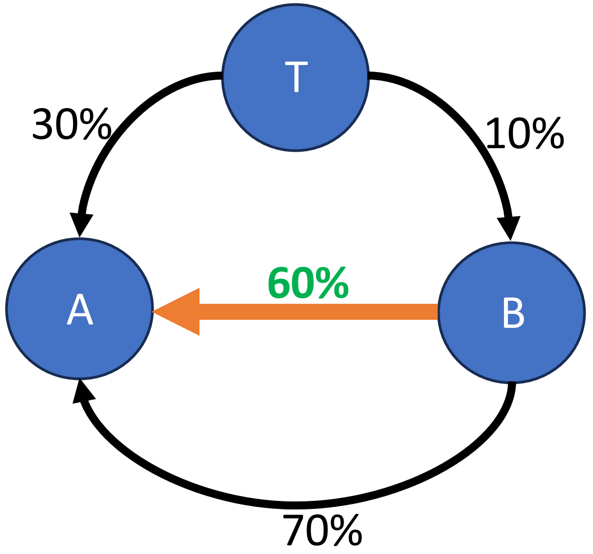
**JBEVal**

- Can process code and text separately or combined
- Threshold $T$ and minimum block length $L_m$ helps to control aggressiveness and adopt
- Can process several files, including previous cohort
- Reveals groups of students colluded together
- is a supportive tool to ***detect potential plagiarism cases***: It allows to find the similar cases and visualisations to trace and manually go through reports **to ensure the plagiarism** for reporting misconduct.

# Future work and development

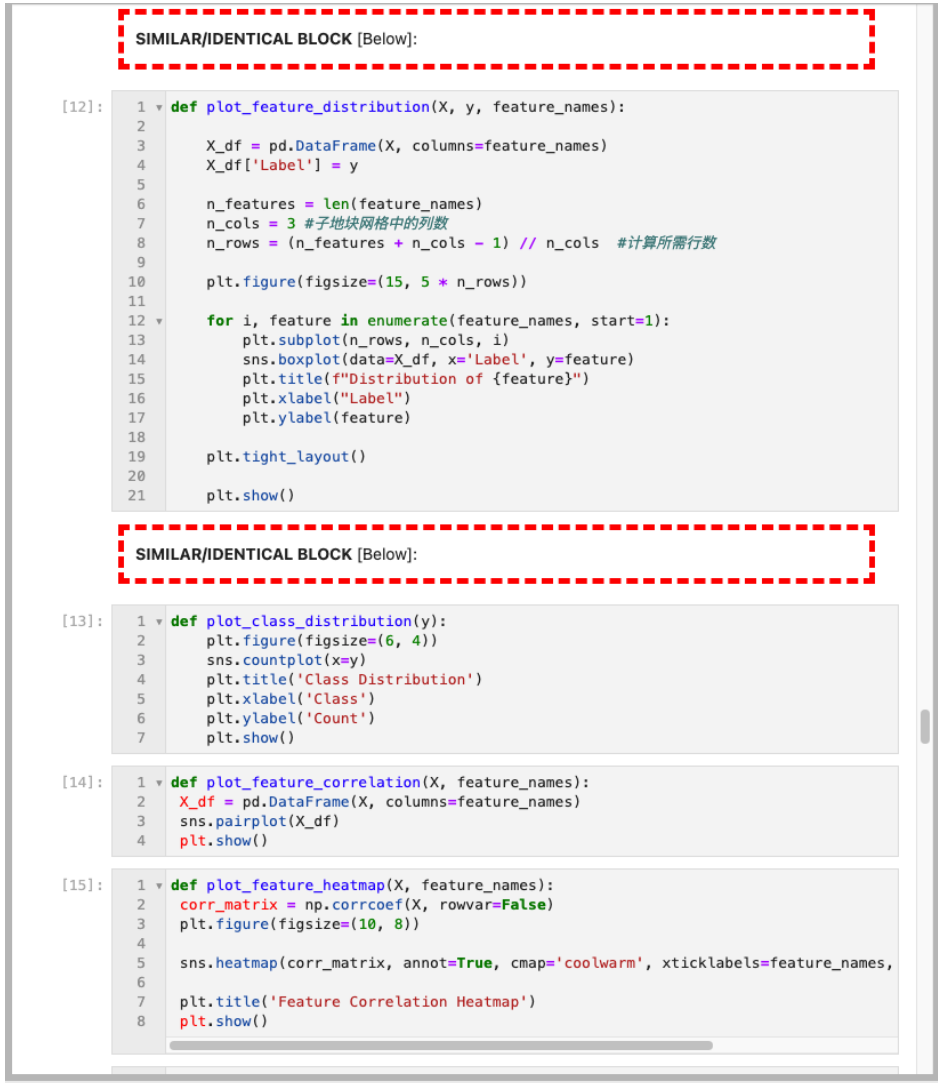
**Working towards:**

- Removing instructions/template components



- Development of an online service for educators to use world-wide
- User friendly interface
- Annotation of jupyter notebooks for similar blocks
- Adapting to different languages (e.g. R)

# Examples: A few cases



**SIMILAR/IDENTICAL BLOCK** [Below]:

[12]:

```python
def plot_feature_distribution(X, y, feature_names):

    X_df = pd.DataFrame(X, columns=feature_names)
    X_df['Label'] = y

    n_features = len(feature_names)
    n_cols = 3 #子地块网格中的列数
    n_rows = (n_features + n_cols - 1) // n_cols  #计算所需行数

    plt.figure(figsize=(15, 5 * n_rows))

    for i, feature in enumerate(feature_names, start=1):
        plt.subplot(n_rows, n_cols, i)
        sns.boxplot(data=X_df, x='Label', y=feature)
        plt.title(f"Distribution of {feature}")
        plt.xlabel("Label")
        plt.ylabel(feature)

    plt.tight_layout()

    plt.show()
```

**SIMILAR/IDENTICAL BLOCK** [Below]:

[13]:

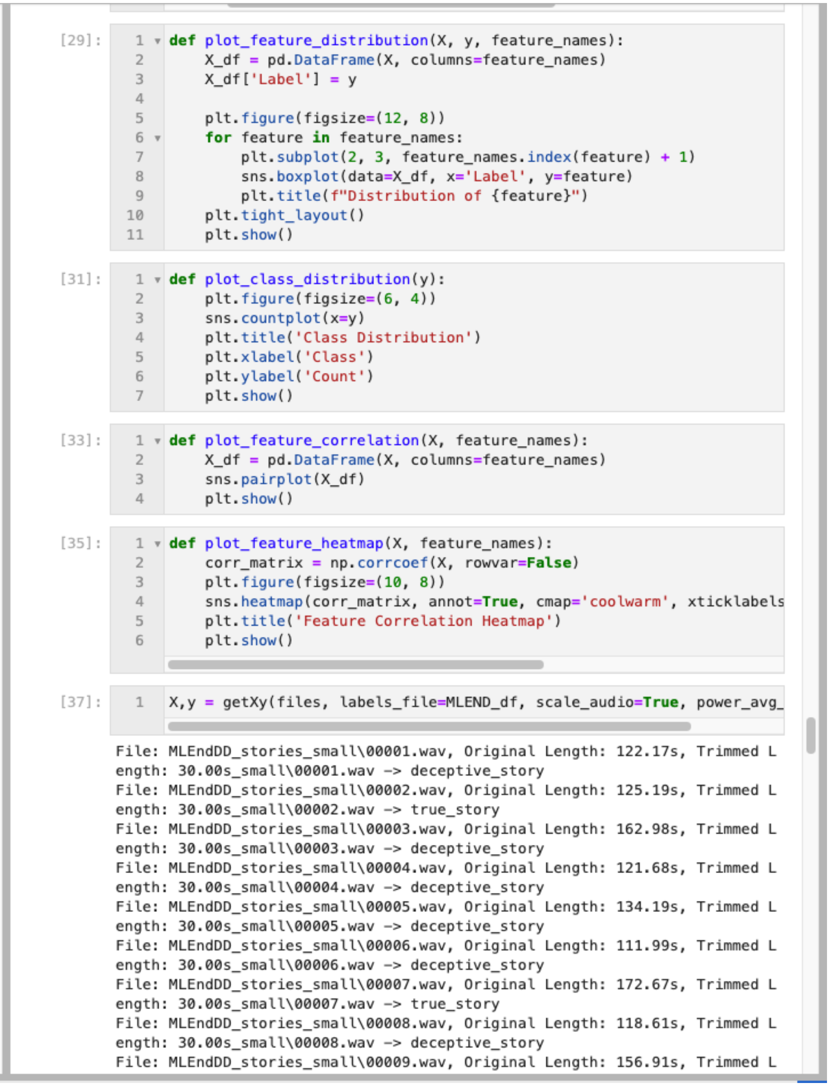
```python
def plot_class_distribution(y):
    plt.figure(figsize=(6, 4))
    sns.countplot(x=y)
    plt.title('Class Distribution')
    plt.xlabel('Class')
    plt.ylabel('Count')
    plt.show()
```

[14]:

```python
def plot_feature_correlation(X, feature_names):
 X_df = pd.DataFrame(X, columns=feature_names)
 sns.pairplot(X_df)
 plt.show()
```

[15]:

```python
def plot_feature_heatmap(X, feature_names):
 corr_matrix = np.corrcoef(X, rowvar=False)
 plt.figure(figsize=(10, 8))

 sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', xticklabels=feature_names,

 plt.title('Feature Correlation Heatmap')
 plt.show()
```



[29]:

```python
def plot_feature_distribution(X, y, feature_names):
    X_df = pd.DataFrame(X, columns=feature_names)
    X_df['Label'] = y

    plt.figure(figsize=(12, 8))
    for feature in feature_names:
        plt.subplot(2, 3, feature_names.index(feature) + 1)
        sns.boxplot(data=X_df, x='Label', y=feature)
        plt.title(f"Distribution of {feature}")
    plt.tight_layout()
    plt.show()
```

[31]:

```python
def plot_class_distribution(y):
    plt.figure(figsize=(6, 4))
    sns.countplot(x=y)
    plt.title('Class Distribution')
    plt.xlabel('Class')
    plt.ylabel('Count')
    plt.show()
```

[33]:

```python
def plot_feature_correlation(X, feature_names):
    X_df = pd.DataFrame(X, columns=feature_names)
    sns.pairplot(X_df)
    plt.show()
```

[35]:

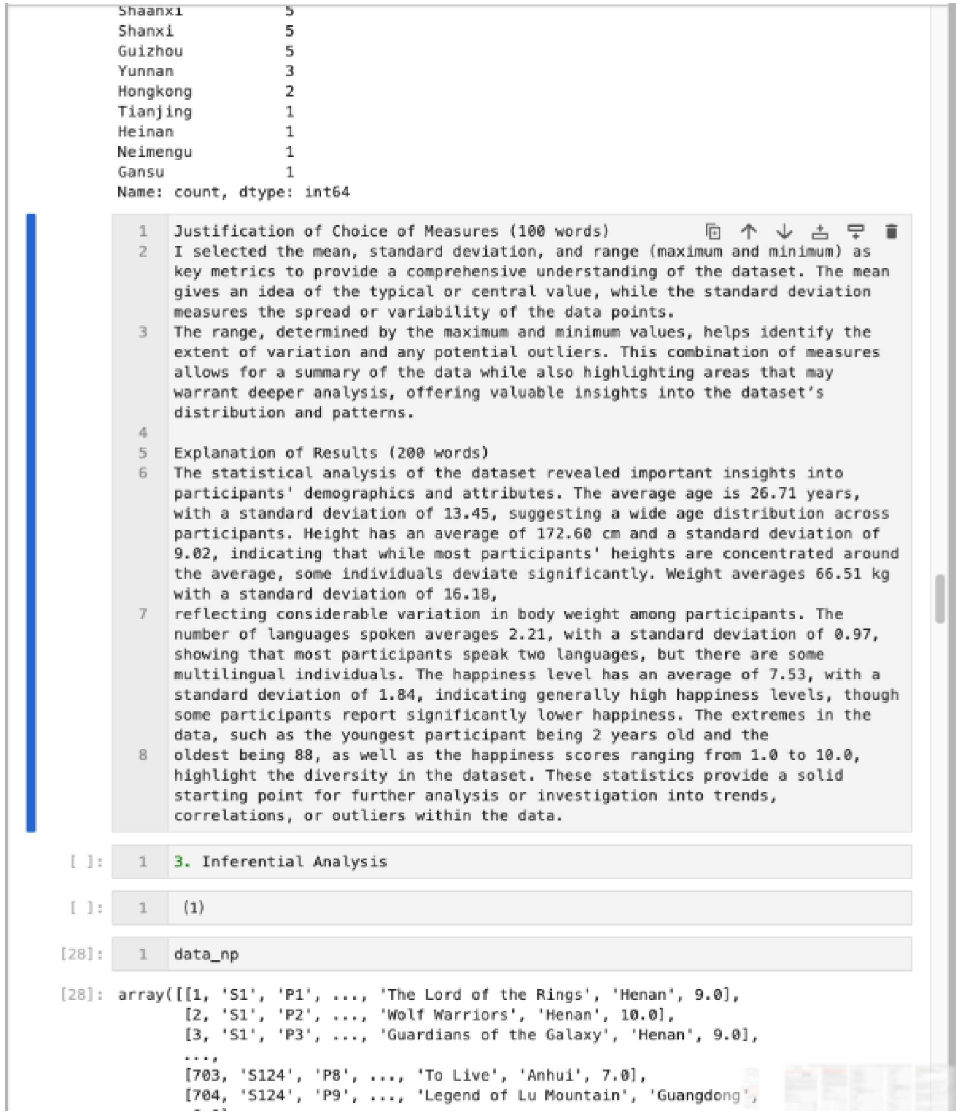
```python
def plot_feature_heatmap(X, feature_names):
    corr_matrix = np.corrcoef(X, rowvar=False)
    plt.figure(figsize=(10, 8))
    sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', xticklabels
    plt.title('Feature Correlation Heatmap')
    plt.show()
```

[37]:

```python
X,y = getXy(files, labels_file=MLEND_df, scale_audio=True, power_avg_
```

```
File: MLEndDD_stories_small\00001.wav, Original Length: 122.17s, Trimmed L
ength: 30.00s_small\00001.wav -> deceptive_story
File: MLEndDD_stories_small\00002.wav, Original Length: 125.19s, Trimmed L
ength: 30.00s_small\00002.wav -> true_story
File: MLEndDD_stories_small\00003.wav, Original Length: 162.98s, Trimmed L
ength: 30.00s_small\00003.wav -> deceptive_story
File: MLEndDD_stories_small\00004.wav, Original Length: 121.68s, Trimmed L
ength: 30.00s_small\00004.wav -> deceptive_story
File: MLEndDD_stories_small\00005.wav, Original Length: 134.19s, Trimmed L
ength: 30.00s_small\00005.wav -> deceptive_story
File: MLEndDD_stories_small\00006.wav, Original Length: 111.99s, Trimmed L
ength: 30.00s_small\00006.wav -> deceptive_story
File: MLEndDD_stories_small\00007.wav, Original Length: 172.67s, Trimmed L
ength: 30.00s_small\00007.wav -> true_story
File: MLEndDD_stories_small\00008.wav, Original Length: 118.61s, Trimmed L
ength: 30.00s_small\00008.wav -> deceptive_story
File: MLEndDD_stories_small\00009.wav, Original Length: 156.91s, Trimmed L
```

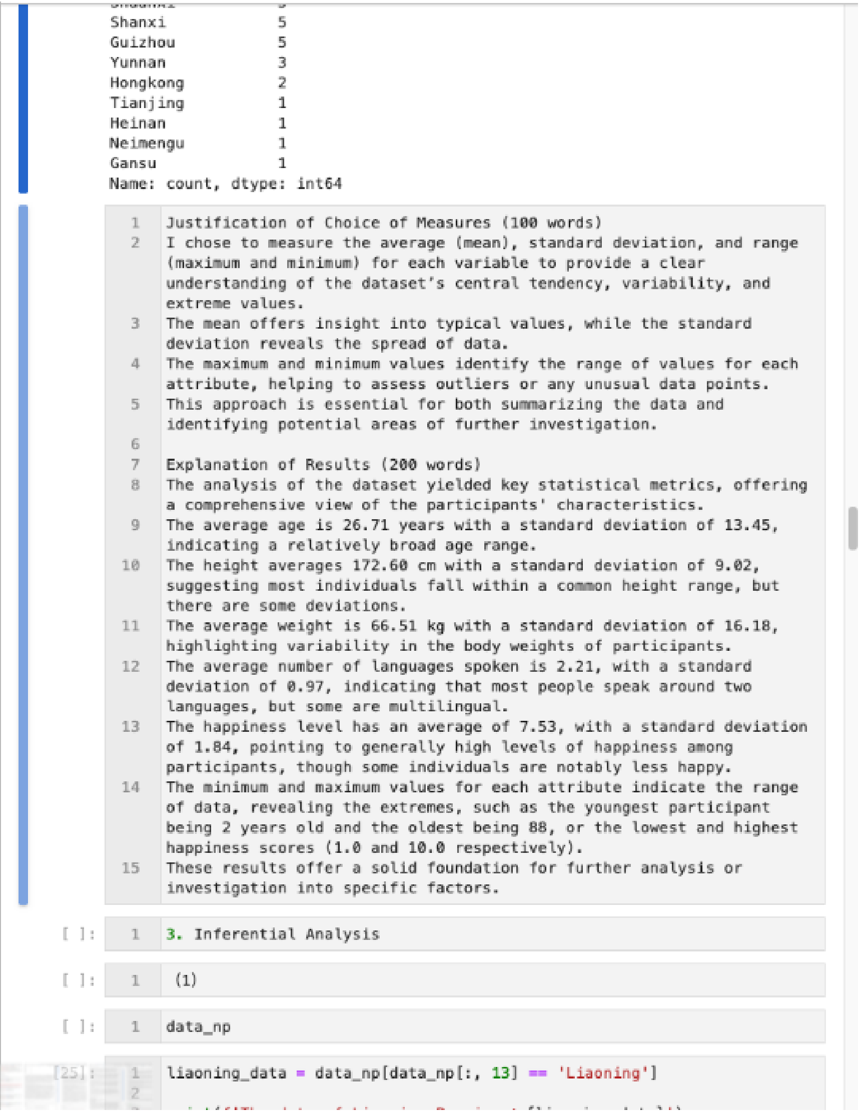
# Examples: A few cases



```
Shaanxi          5
Shanxi           5
Guizhou          5
Yunnan           3
Hongkong         2
Tianjing         1
Heinan           1
Neimengu         1
Gansu            1
Name: count, dtype: int64
```

```
Justification of Choice of Measures (100 words)
I selected the mean, standard deviation, and range (maximum and minimum) as
key metrics to provide a comprehensive understanding of the dataset. The mean
gives an idea of the typical or central value, while the standard deviation
measures the spread or variability of the data points.
The range, determined by the maximum and minimum values, helps identify the
extent of variation and any potential outliers. This combination of measures
allows for a summary of the data while also highlighting areas that may
warrant deeper analysis, offering valuable insights into the dataset's
distribution and patterns.

Explanation of Results (200 words)
The statistical analysis of the dataset revealed important insights into
participants' demographics and attributes. The average age is 26.71 years,
with a standard deviation of 13.45, suggesting a wide age distribution across
participants. Height has an average of 172.60 cm and a standard deviation of
9.02, indicating that while most participants' heights are concentrated around
the average, some individuals deviate significantly. Weight averages 66.51 kg
with a standard deviation of 16.18,
reflecting considerable variation in body weight among participants. The
number of languages spoken averages 2.21, with a standard deviation of 0.97,
showing that most participants speak two languages, but there are some
multilingual individuals. The happiness level has an average of 7.53, with a
standard deviation of 1.84, indicating generally high happiness levels, though
some participants report significantly lower happiness. The extremes in the
data, such as the youngest participant being 2 years old and the
oldest being 88, as well as the happiness scores ranging from 1.0 to 10.0,
highlight the diversity in the dataset. These statistics provide a solid
starting point for further analysis or investigation into trends,
correlations, or outliers within the data.
```

```
[ ]: 3. Inferential Analysis

[ ]:  (1)

[28]: data_np

[28]: array([[1, 'S1', 'P1', ..., 'The Lord of the Rings', 'Henan', 9.0],
             [2, 'S1', 'P2', ..., 'Wolf Warriors', 'Henan', 10.0],
             [3, 'S1', 'P3', ..., 'Guardians of the Galaxy', 'Henan', 9.0],
             ...,
             [703, 'S124', 'P8', ..., 'To Live', 'Anhui', 7.0],
             [704, 'S124', 'P9', ..., 'Legend of Lu Mountain', 'Guangdong',
```

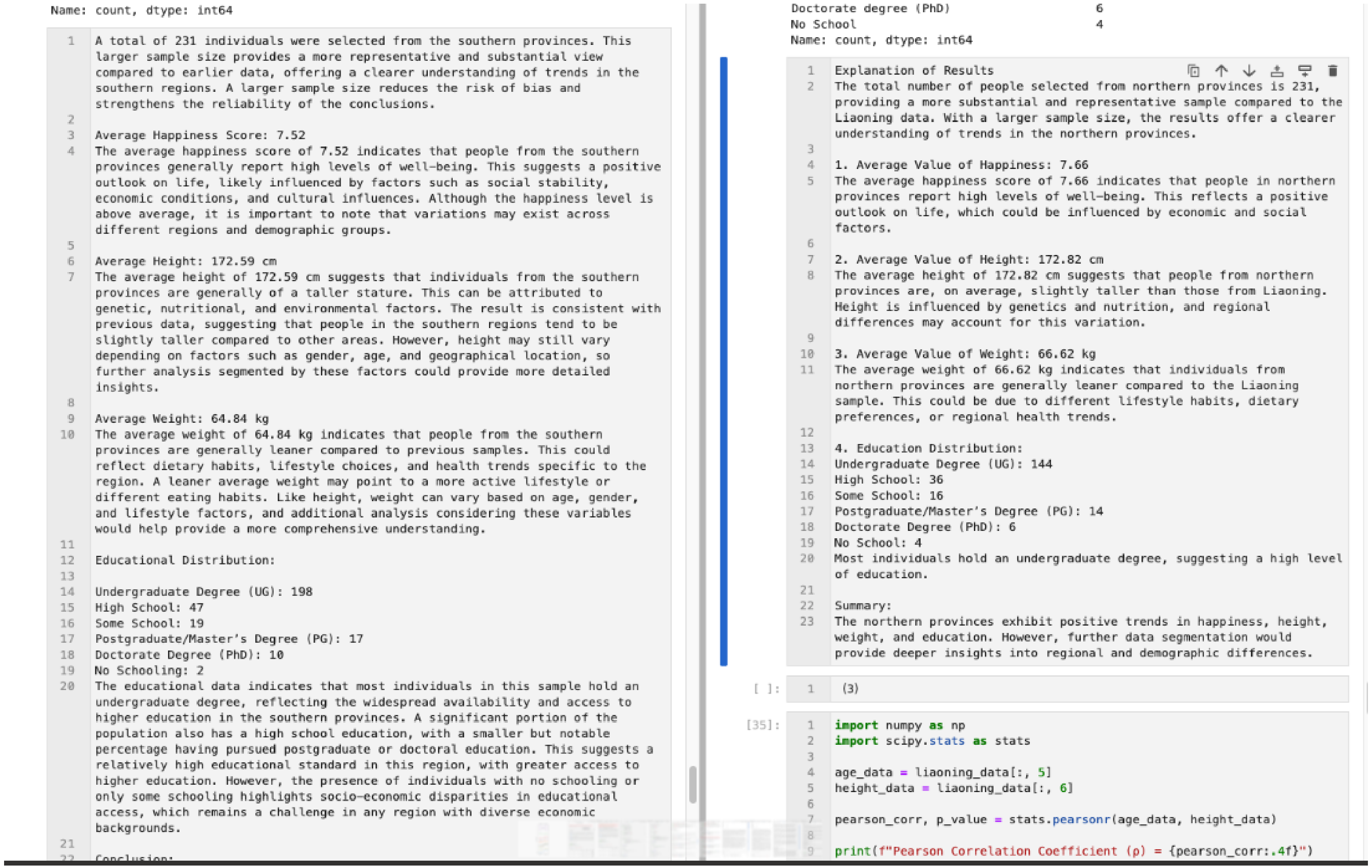


```
Shanxi           5
Guizhou          5
Yunnan           3
Hongkong         2
Tianjing         1
Heinan           1
Neimengu         1
Gansu            1
Name: count, dtype: int64
```

```
Justification of Choice of Measures (100 words)
I chose to measure the average (mean), standard deviation, and range
(maximum and minimum) for each variable to provide a clear
understanding of the dataset's central tendency, variability, and
extreme values.
The mean offers insight into typical values, while the standard
deviation reveals the spread of data.
The maximum and minimum values identify the range of values for each
attribute, helping to assess outliers or any unusual data points.
This approach is essential for both summarizing the data and
identifying potential areas of further investigation.

Explanation of Results (200 words)
The analysis of the dataset yielded key statistical metrics, offering
a comprehensive view of the participants' characteristics.
The average age is 26.71 years with a standard deviation of 13.45,
indicating a relatively broad age range.
The height averages 172.60 cm with a standard deviation of 9.02,
suggesting most individuals fall within a common height range, but
there are some deviations.
The average weight is 66.51 kg with a standard deviation of 16.18,
highlighting variability in the body weights of participants.
The average number of languages spoken is 2.21, with a standard
deviation of 0.97, indicating that most people speak around two
languages, but some are multilingual.
The happiness level has an average of 7.53, with a standard deviation
of 1.84, pointing to generally high levels of happiness among
participants, though some individuals are notably less happy.
The minimum and maximum values for each attribute indicate the range
of data, revealing the extremes, such as the youngest participant
being 2 years old and the oldest being 88, or the lowest and highest
happiness scores (1.0 and 10.0 respectively).
These results offer a solid foundation for further analysis or
investigation into specific factors.
```

```
[ ]: 3. Inferential Analysis

[ ]:  (1)

[ ]: data_np

[25]: liaoning_data = data_np[data_np[:, 13] == 'Liaoning']
```

# Examples: A few cases



```
Name: count, dtype: int64
```

A total of 231 individuals were selected from the southern provinces. This larger sample size provides a more representative and substantial view compared to earlier data, offering a clearer understanding of trends in the southern regions. A larger sample size reduces the risk of bias and strengthens the reliability of the conclusions.

Average Happiness Score: 7.52
The average happiness score of 7.52 indicates that people from the southern provinces generally report high levels of well-being. This suggests a positive outlook on life, likely influenced by factors such as social stability, economic conditions, and cultural influences. Although the happiness level is above average, it is important to note that variations may exist across different regions and demographic groups.

Average Height: 172.59 cm
The average height of 172.59 cm suggests that individuals from the southern provinces are generally of a taller stature. This can be attributed to genetic, nutritional, and environmental factors. The result is consistent with previous data, suggesting that people in the southern regions tend to be slightly taller compared to other areas. However, height may still vary depending on factors such as gender, age, and geographical location, so further analysis segmented by these factors could provide more detailed insights.

Average Weight: 64.84 kg
The average weight of 64.84 kg indicates that people from the southern provinces are generally leaner compared to previous samples. This could reflect dietary habits, lifestyle choices, and health trends specific to the region. A leaner average weight may point to a more active lifestyle or different eating habits. Like height, weight can vary based on age, gender, and lifestyle factors, and additional analysis considering these variables would help provide a more comprehensive understanding.

Educational Distribution:

Undergraduate Degree (UG): 198
High School: 47
Some School: 19
Postgraduate/Master’s Degree (PG): 17
Doctorate Degree (PhD): 10
No Schooling: 2
The educational data indicates that most individuals in this sample hold an undergraduate degree, reflecting the widespread availability and access to higher education in the southern provinces. A significant portion of the population also has a high school education, with a smaller but notable percentage having pursued postgraduate or doctoral education. This suggests a relatively high educational standard in this region, with greater access to higher education. However, the presence of individuals with no schooling or only some schooling highlights socio-economic disparities in educational access, which remains a challenge in any region with diverse economic backgrounds.

Conclusion:

```
Doctorate degree (PhD)              6
No School                           4
Name: count, dtype: int64
```

Explanation of Results
The total number of people selected from northern provinces is 231, providing a more substantial and representative sample compared to the Liaoning data. With a larger sample size, the results offer a clearer understanding of trends in the northern provinces.

1. Average Value of Happiness: 7.66
The average happiness score of 7.66 indicates that people in northern provinces report high levels of well-being. This reflects a positive outlook on life, which could be influenced by economic and social factors.

2. Average Value of Height: 172.82 cm
The average height of 172.82 cm suggests that people from northern provinces are, on average, slightly taller than those from Liaoning. Height is influenced by genetics and nutrition, and regional differences may account for this variation.

3. Average Value of Weight: 66.62 kg
The average weight of 66.62 kg indicates that individuals from northern provinces are generally leaner compared to the Liaoning sample. This could be due to different lifestyle habits, dietary preferences, or regional health trends.

4. Education Distribution:
Undergraduate Degree (UG): 144
High School: 36
Some School: 16
Postgraduate/Master’s Degree (PG): 14
Doctorate Degree (PhD): 6
No School: 4
Most individuals hold an undergraduate degree, suggesting a high level of education.

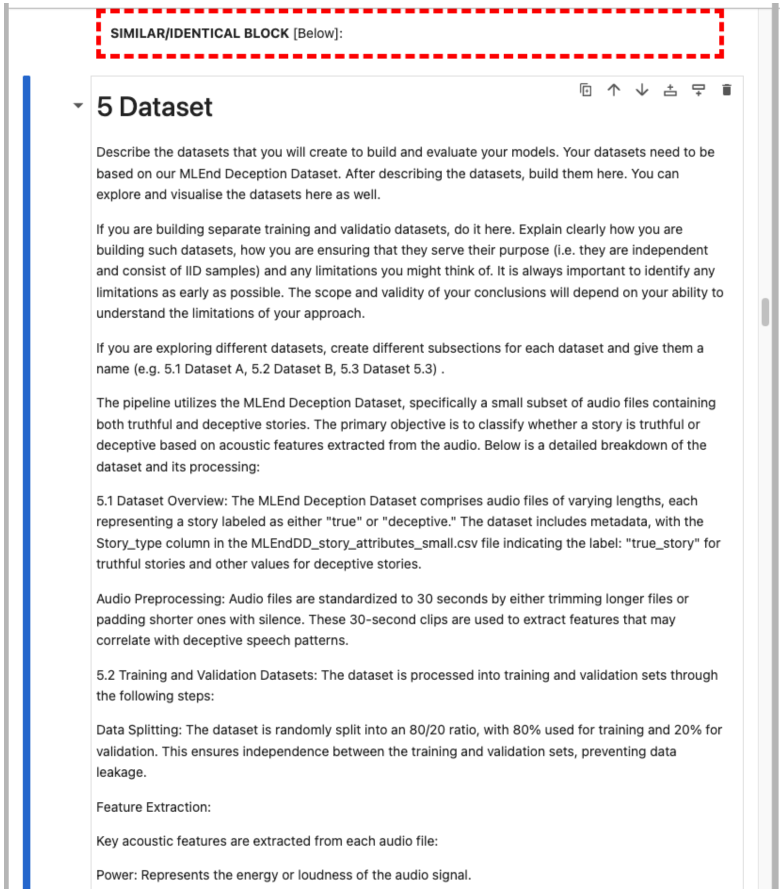
Summary:
The northern provinces exhibit positive trends in happiness, height, weight, and education. However, further data segmentation would provide deeper insights into regional and demographic differences.

[ ]: (3)

[35]:

```python
import numpy as np
import scipy.stats as stats

age_data = liaoning_data[:, 5]
height_data = liaoning_data[:, 6]

pearson_corr, p_value = stats.pearsonr(age_data, height_data)

print(f"Pearson Correlation Coefficient (ρ) = {pearson_corr:.4f}")
```

# Examples: A few cases



SIMILAR/IDENTICAL BLOCK [Below]:

## 5 Dataset

Describe the datasets that you will create to build and evaluate your models. Your datasets need to be based on our MLEnd Deception Dataset. After describing the datasets, build them here. You can explore and visualise the datasets here as well.

If you are building separate training and validatio datasets, do it here. Explain clearly how you are building such datasets, how you are ensuring that they serve their purpose (i.e. they are independent and consist of IID samples) and any limitations you might think of. It is always important to identify any limitations as early as possible. The scope and validity of your conclusions will depend on your ability to understand the limitations of your approach.

If you are exploring different datasets, create different subsections for each dataset and give them a name (e.g. 5.1 Dataset A, 5.2 Dataset B, 5.3 Dataset 5.3) .

The pipeline utilizes the MLEnd Deception Dataset, specifically a small subset of audio files containing both truthful and deceptive stories. The primary objective is to classify whether a story is truthful or deceptive based on acoustic features extracted from the audio. Below is a detailed breakdown of the dataset and its processing:

5.1 Dataset Overview: The MLEnd Deception Dataset comprises audio files of varying lengths, each representing a story labeled as either "true" or "deceptive." The dataset includes metadata, with the Story_type column in the MLEndDD_story_attributes_small.csv file indicating the label: "true_story" for truthful stories and other values for deceptive stories.

Audio Preprocessing: Audio files are standardized to 30 seconds by either trimming longer files or padding shorter ones with silence. These 30-second clips are used to extract features that may correlate with deceptive speech patterns.

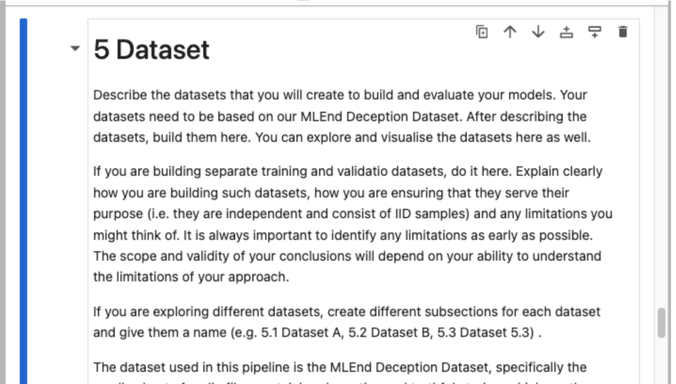
5.2 Training and Validation Datasets: The dataset is processed into training and validation sets through the following steps:

Data Splitting: The dataset is randomly split into an 80/20 ratio, with 80% used for training and 20% for validation. This ensures independence between the training and validation sets, preventing data leakage.

Feature Extraction:

Key acoustic features are extracted from each audio file:

Power: Represents the energy or loudness of the audio signal.



## 5 Dataset

Describe the datasets that you will create to build and evaluate your models. Your datasets need to be based on our MLEnd Deception Dataset. After describing the datasets, build them here. You can explore and visualise the datasets here as well.

If you are building separate training and validatio datasets, do it here. Explain clearly how you are building such datasets, how you are ensuring that they serve their purpose (i.e. they are independent and consist of IID samples) and any limitations you might think of. It is always important to identify any limitations as early as possible. The scope and validity of your conclusions will depend on your ability to understand the limitations of your approach.

If you are exploring different datasets, create different subsections for each dataset and give them a name (e.g. 5.1 Dataset A, 5.2 Dataset B, 5.3 Dataset 5.3) .

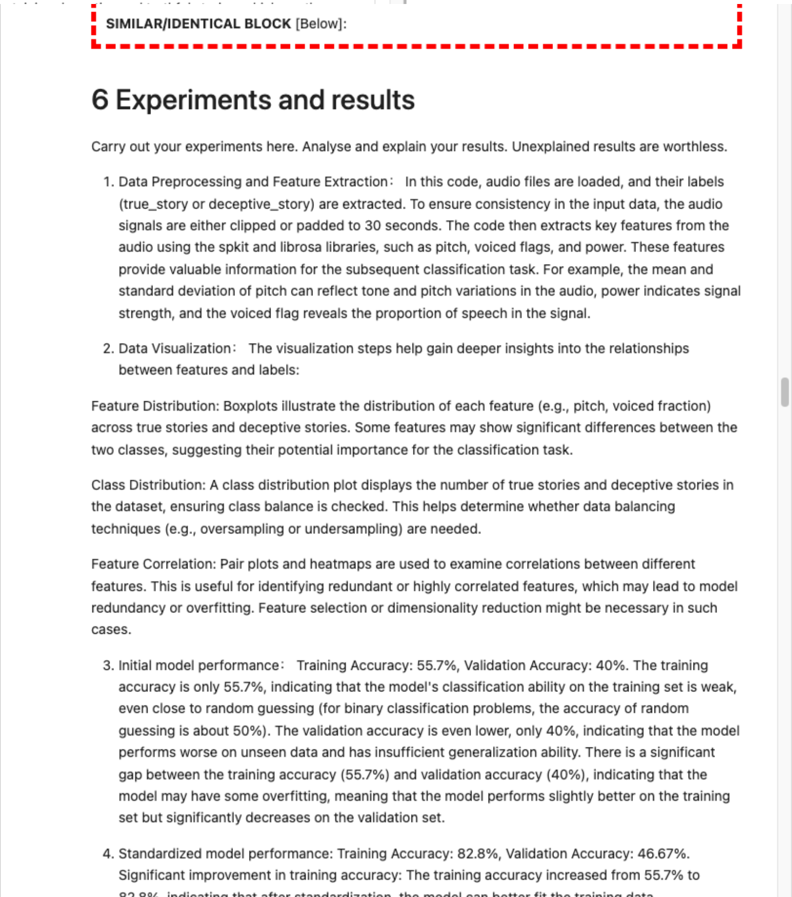
The dataset used in this pipeline is the MLEnd Deception Dataset, specifically the
small subset of audio files c
basis for the classification ta
audio files along with corres
a true story or a deceptive s

5.1 Dataset Overview The M
lengths. Each audio file repr
includes labels indicating wh
goal is to train a model to pr
power, and voiced fraction.

In this specific case, the auc
files or padding shorter files
extract features relevant to
labels for these audio files a
MLEndDD_story_attributes_
the value "true_story" repre
deceptive story.

5.2 Training and Validation D
the MLEnd Deception Datas

Data Splitting: The audio file
the other for validating it. Th
the data is used for training
randomly to ensure the inde



SIMILAR/IDENTICAL BLOCK [Below]:

## 6 Experiments and results

Carry out your experiments here. Analyse and explain your results. Unexplained results are worthless.

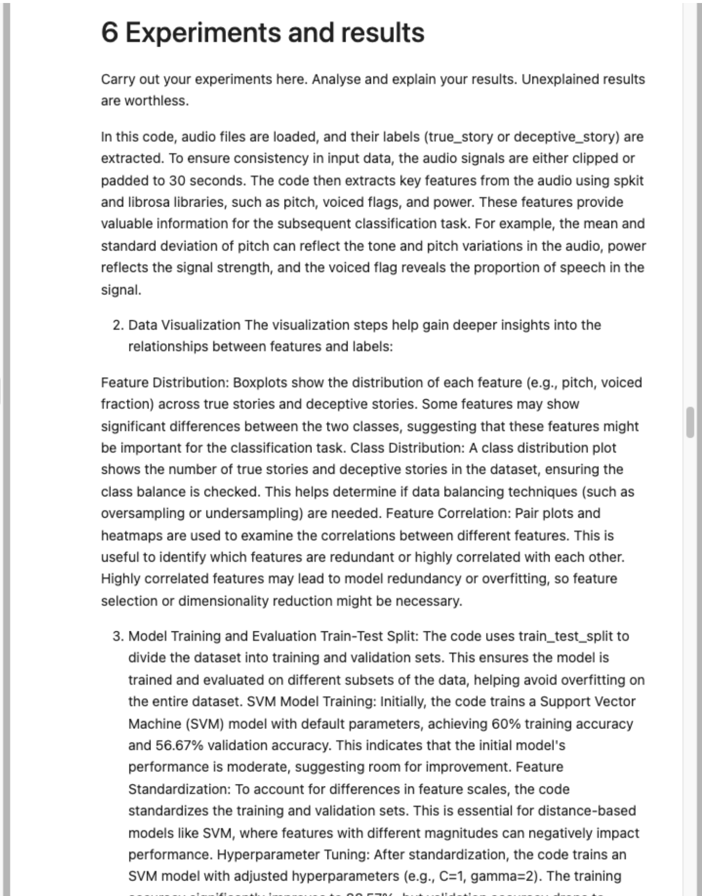
1. Data Preprocessing and Feature Extraction： In this code, audio files are loaded, and their labels (true_story or deceptive_story) are extracted. To ensure consistency in the input data, the audio signals are either clipped or padded to 30 seconds. The code then extracts key features from the audio using the spkit and librosa libraries, such as pitch, voiced flags, and power. These features provide valuable information for the subsequent classification task. For example, the mean and standard deviation of pitch can reflect tone and pitch variations in the audio, power indicates signal strength, and the voiced flag reveals the proportion of speech in the signal.
2. Data Visualization： The visualization steps help gain deeper insights into the relationships between features and labels:

Feature Distribution: Boxplots illustrate the distribution of each feature (e.g., pitch, voiced fraction) across true stories and deceptive stories. Some features may show significant differences between the two classes, suggesting their potential importance for the classification task.

Class Distribution: A class distribution plot displays the number of true stories and deceptive stories in the dataset, ensuring class balance is checked. This helps determine whether data balancing techniques (e.g., oversampling or undersampling) are needed.

Feature Correlation: Pair plots and heatmaps are used to examine correlations between different features. This is useful for identifying redundant or highly correlated features, which may lead to model redundancy or overfitting. Feature selection or dimensionality reduction might be necessary in such cases.

3. Initial model performance： Training Accuracy: 55.7%, Validation Accuracy: 40%. The training accuracy is only 55.7%, indicating that the model's classification ability on the training set is weak, even close to random guessing (for binary classification problems, the accuracy of random guessing is about 50%). The validation accuracy is even lower, only 40%, indicating that the model performs worse on unseen data and has insufficient generalization ability. There is a significant gap between the training accuracy (55.7%) and validation accuracy (40%), indicating that the model may have some overfitting, meaning that the model performs slightly better on the training set but significantly decreases on the validation set.
4. Standardized model performance: Training Accuracy: 82.8%, Validation Accuracy: 46.67%. Significant improvement in training accuracy: The training accuracy increased from 55.7% to 82.8%, indicating that after standardization, the model can better fit the training data.



## 6 Experiments and results

Carry out your experiments here. Analyse and explain your results. Unexplained results are worthless.

In this code, audio files are loaded, and their labels (true_story or deceptive_story) are extracted. To ensure consistency in input data, the audio signals are either clipped or padded to 30 seconds. The code then extracts key features from the audio using spkit and librosa libraries, such as pitch, voiced flags, and power. These features provide valuable information for the subsequent classification task. For example, the mean and standard deviation of pitch can reflect the tone and pitch variations in the audio, power reflects the signal strength, and the voiced flag reveals the proportion of speech in the signal.

2. Data Visualization The visualization steps help gain deeper insights into the relationships between features and labels:

Feature Distribution: Boxplots show the distribution of each feature (e.g., pitch, voiced fraction) across true stories and deceptive stories. Some features may show significant differences between the two classes, suggesting that these features might be important for the classification task. Class Distribution: A class distribution plot shows the number of true stories and deceptive stories in the dataset, ensuring the class balance is checked. This helps determine if data balancing techniques (such as oversampling or undersampling) are needed. Feature Correlation: Pair plots and heatmaps are used to examine the correlations between different features. This is useful to identify which features are redundant or highly correlated with each other. Highly correlated features may lead to model redundancy or overfitting, so feature selection or dimensionality reduction might be necessary.

3. Model Training and Evaluation Train-Test Split: The code uses train_test_split to divide the dataset into training and validation sets. This ensures the model is trained and evaluated on different subsets of the data, helping avoid overfitting on the entire dataset. SVM Model Training: Initially, the code trains a Support Vector Machine (SVM) model with default parameters, achieving 60% training accuracy and 56.67% validation accuracy. This indicates that the initial model's performance is moderate, suggesting room for improvement. Feature Standardization: To account for differences in feature scales, the code standardizes the training and validation sets. This is essential for distance-based models like SVM, where features with different magnitudes can negatively impact performance. Hyperparameter Tuning: After standardization, the code trains an SVM model with adjusted hyperparameters (e.g., C=1, gamma=2). The training accuracy significantly improves to 83.57%, but validation accuracy drops to

IEEE EDUCON 2025
IEEE Global Engineering Education Conference
London, United Kingdom || 22-25 April 2025 || Queen Mary University of London

# A Tool for Detecting Similarities in Jupyter Notebooks Used as Assessment Reports

## Team

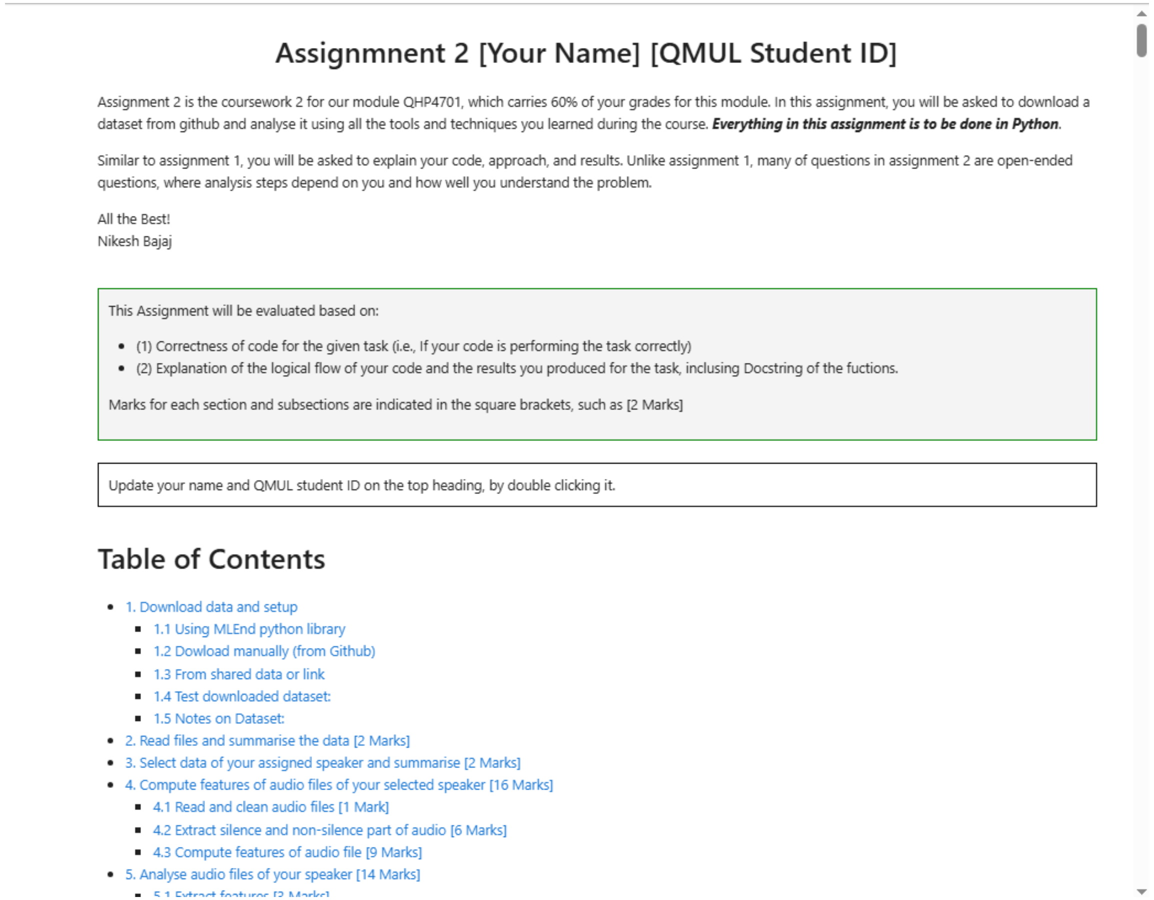
**Nikesh Bajaj, Dimitrios Chiotis, Reza Moosaei, Jordan B. L. Smith, Pengfei Fan, Jesús Requena Carrión**

# Any Question?

## Feedback/Suggestions @

**Nikesh Bajaj**
School of Physical and Chemical Sciences
Queen Mary School Hainan
nikesh.bajaj@qmul.ac.uk
https://nikeshbajaj.in

# Examples of Jupyter Notebooks as assessments



## Assignmnent 2 [Your Name] [QMUL Student ID]

Assignment 2 is the coursework 2 for our module QHP4701, which carries 60% of your grades for this module. In this assignment, you will be asked to download a dataset from github and analyse it using all the tools and techniques you learned during the course. ***Everything in this assignment is to be done in Python***.

Similar to assignment 1, you will be asked to explain your code, approach, and results. Unlike assignment 1, many of questions in assignment 2 are open-ended questions, where analysis steps depend on you and how well you understand the problem.

All the Best!
Nikesh Bajaj

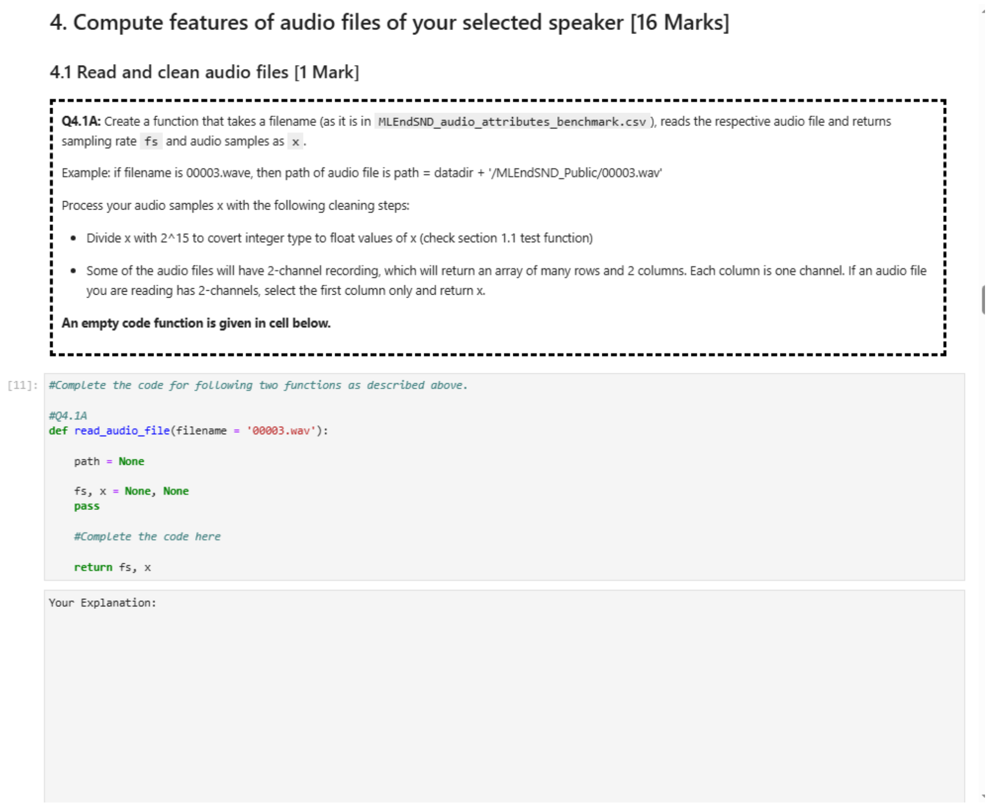
This Assignment will be evaluated based on:

- (1) Correctness of code for the given task (i.e., If your code is performing the task correctly)
- (2) Explanation of the logical flow of your code and the results you produced for the task, inclusing Docstring of the fuctions.

Marks for each section and subsections are indicated in the square brackets, such as [2 Marks]

Update your name and QMUL student ID on the top heading, by double clicking it.

## Table of Contents



## 4. Compute features of audio files of your selected speaker [16 Marks]

### 4.1 Read and clean audio files [1 Mark]

**Q4.1A:** Create a function that takes a filename (as it is in `MLEndSND_audio_attributes_benchmark.csv`), reads the respective audio file and returns sampling rate `fs` and audio samples as `x`.

Example: if filename is 00003.wave, then path of audio file is path = datadir + '/MLEndSND_Public/00003.wav'

Process your audio samples x with the following cleaning steps:

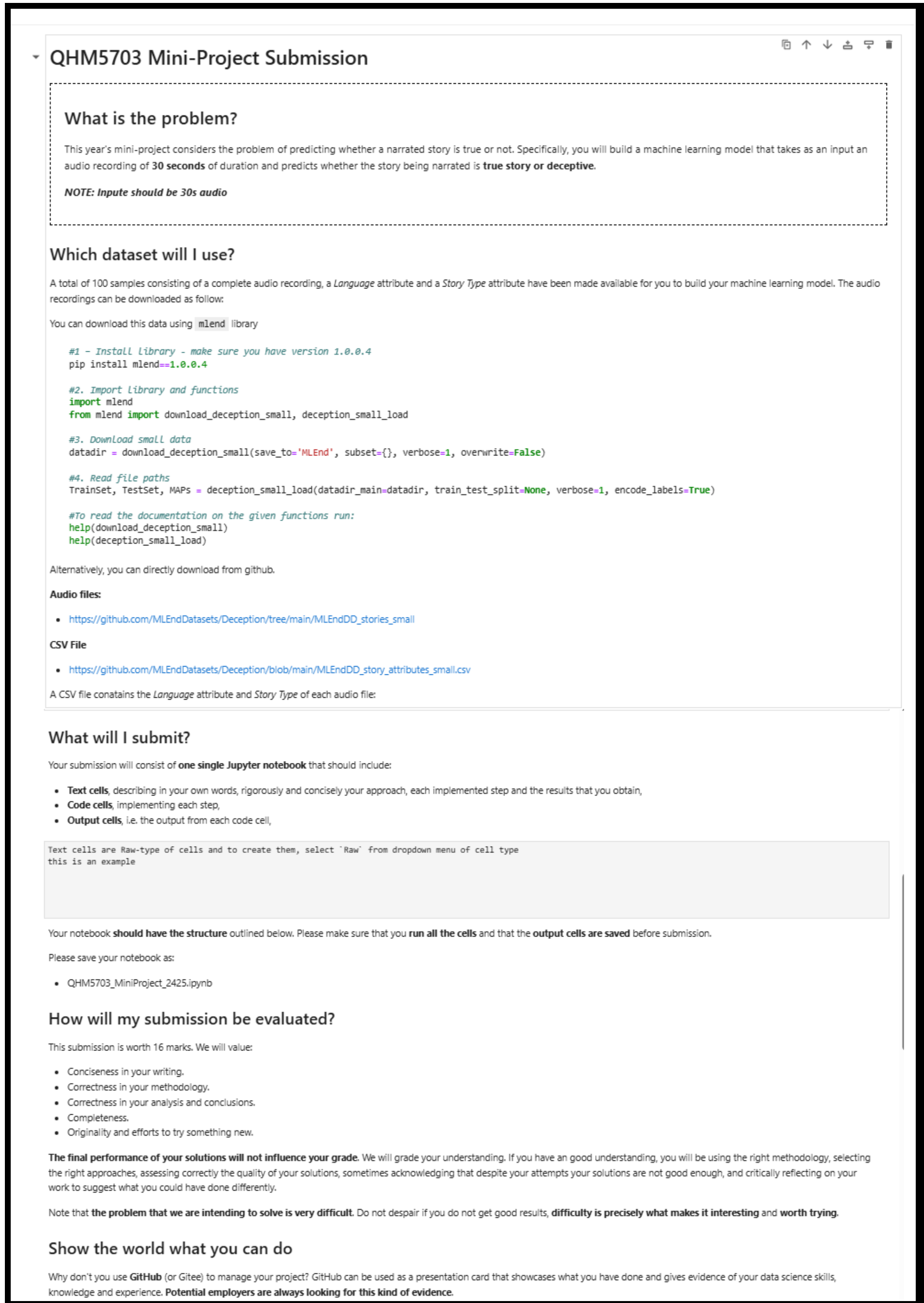
- Divide x with 2^15 to covert integer type to float values of x (check section 1.1 test function)
- Some of the audio files will have 2-channel recording, which will return an array of many rows and 2 columns. Each column is one channel. If an audio file you are reading has 2-channels, select the first column only and return x.

**An empty code function is given in cell below.**

```
[11]: #Complete the code for following two functions as described above.

      #Q4.1A
      def read_audio_file(filename = '00003.wav'):

          path = None

          fs, x = None, None
          pass

          #Complete the code here

          return fs, x
```

```
Your Explanation:
```

# Examples of Jupyter Notebooks as assessments



## QHM5703 Mini-Project Submission

### What is the problem?

This year's mini-project considers the problem of predicting whether a narrated story is true or not. Specifically, you will build a machine learning model that takes as an input an audio recording of **30 seconds** of duration and predicts whether the story being narrated is **true story or deceptive**.

***NOTE: Inpute should be 30s audio***

### Which dataset will I use?

A total of 100 samples consisting of a complete audio recording, a *Language* attribute and a *Story Type* attribute have been made available for you to build your machine learning model. The audio recordings can be downloaded as follow:

You can download this data using `mlend` library

```
#1 - Install library - make sure you have version 1.0.0.4
pip install mlend==1.0.0.4

#2. Import library and functions
import mlend
from mlend import download_deception_small, deception_small_load

#3. Download small data
datadir = download_deception_small(save_to='MLEnd', subset={}, verbose=1, overwrite=False)

#4. Read file paths
TrainSet, TestSet, MAPs = deception_small_load(datadir_main=datadir, train_test_split=None, verbose=1, encode_labels=True)

#To read the documentation on the given functions run:
help(download_deception_small)
help(deception_small_load)
```

Alternatively, you can directly download from github.

**Audio files:**

- https://github.com/MLEndDatasets/Deception/tree/main/MLEndDD_stories_small

**CSV File**

- https://github.com/MLEndDatasets/Deception/blob/main/MLEndDD_story_attributes_small.csv

A CSV file conatains the *Language* attribute and *Story Type* of each audio file:

### What will I submit?

Your submission will consist of **one single Jupyter notebook** that should include:

- **Text cells**, describing in your own words, rigorously and concisely your approach, each implemented step and the results that you obtain,
- **Code cells**, implementing each step,
- **Output cells**, i.e. the output from each code cell,

```
Text cells are Raw-type of cells and to create them, select `Raw` from dropdown menu of cell type
this is an example
```

Your notebook **should have the structure** outlined below. Please make sure that you **run all the cells** and that the **output cells are saved** before submission.

Please save your notebook as:

- QHM5703_MiniProject_2425.ipynb

### How will my submission be evaluated?

This submission is worth 16 marks. We will value:

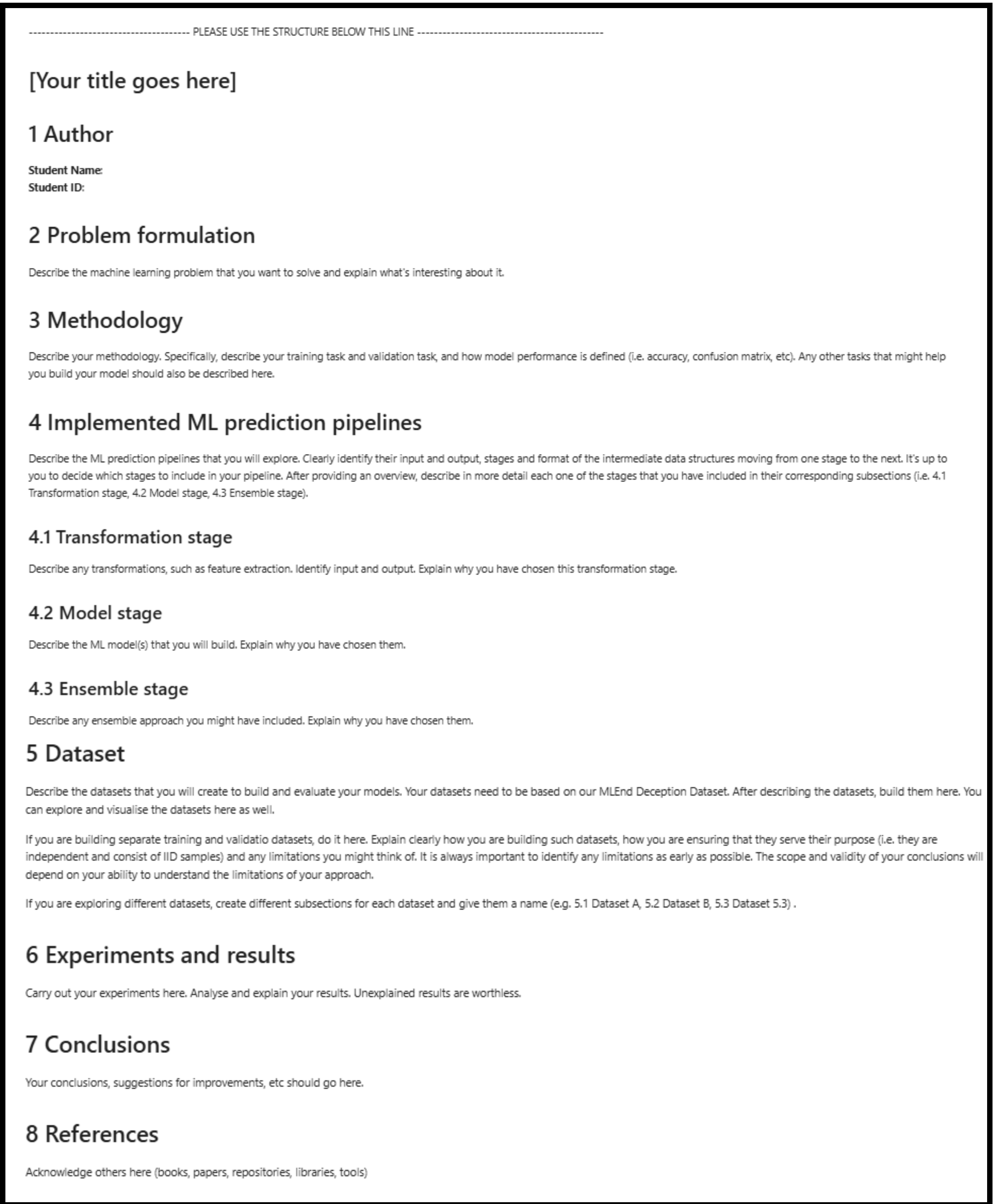
- Conciseness in your writing.
- Correctness in your methodology.
- Correctness in your analysis and conclusions.
- Completeness.
- Originality and efforts to try something new.

**The final performance of your solutions will not influence your grade.** We will grade your understanding. If you have an good understanding, you will be using the right methodology, selecting the right approaches, assessing correctly the quality of your solutions, sometimes acknowledging that despite your attempts your solutions are not good enough, and critically reflecting on your work to suggest what you could have done differently.

Note that **the problem that we are intending to solve is very difficult.** Do not despair if you do not get good results, **difficulty is precisely what makes it interesting** and **worth trying.**

### Show the world what you can do

Why don't you use **GitHub** (or Gitee) to manage your project? GitHub can be used as a presentation card that showcases what you have done and gives evidence of your data science skills, knowledge and experience. **Potential employers are always looking for this kind of evidence.**



-------------------------------------------- PLEASE USE THE STRUCTURE BELOW THIS LINE --------------------------------------------

## [Your title goes here]

## 1 Author

**Student Name:**
**Student ID:**

## 2 Problem formulation

Describe the machine learning problem that you want to solve and explain what's interesting about it.

## 3 Methodology

Describe your methodology. Specifically, describe your training task and validation task, and how model performance is defined (i.e. accuracy, confusion matrix, etc). Any other tasks that might help you build your model should also be described here.

## 4 Implemented ML prediction pipelines

Describe the ML prediction pipelines that you will explore. Clearly identify their input and output, stages and format of the intermediate data structures moving from one stage to the next. It's up to you to decide which stages to include in your pipeline. After providing an overview, describe in more detail each one of the stages that you have included in their corresponding subsections (i.e. 4.1 Transformation stage, 4.2 Model stage, 4.3 Ensemble stage).

### 4.1 Transformation stage

Describe any transformations, such as feature extraction. Identify input and output. Explain why you have chosen this transformation stage.

### 4.2 Model stage

Describe the ML model(s) that you will build. Explain why you have chosen them.

### 4.3 Ensemble stage

Describe any ensemble approach you might have included. Explain why you have chosen them.

## 5 Dataset

Describe the datasets that you will create to build and evaluate your models. Your datasets need to be based on our MLEnd Deception Dataset. After describing the datasets, build them here. You can explore and visualise the datasets here as well.

If you are building separate training and validatio datasets, do it here. Explain clearly how you are building such datasets, how you are ensuring that they serve their purpose (i.e. they are independent and consist of IID samples) and any limitations you might think of. It is always important to identify any limitations as early as possible. The scope and validity of your conclusions will depend on your ability to understand the limitations of your approach.

If you are exploring different datasets, create different subsections for each dataset and give them a name (e.g. 5.1 Dataset A, 5.2 Dataset B, 5.3 Dataset 5.3) .

## 6 Experiments and results

Carry out your experiments here. Analyse and explain your results. Unexplained results are worthless.

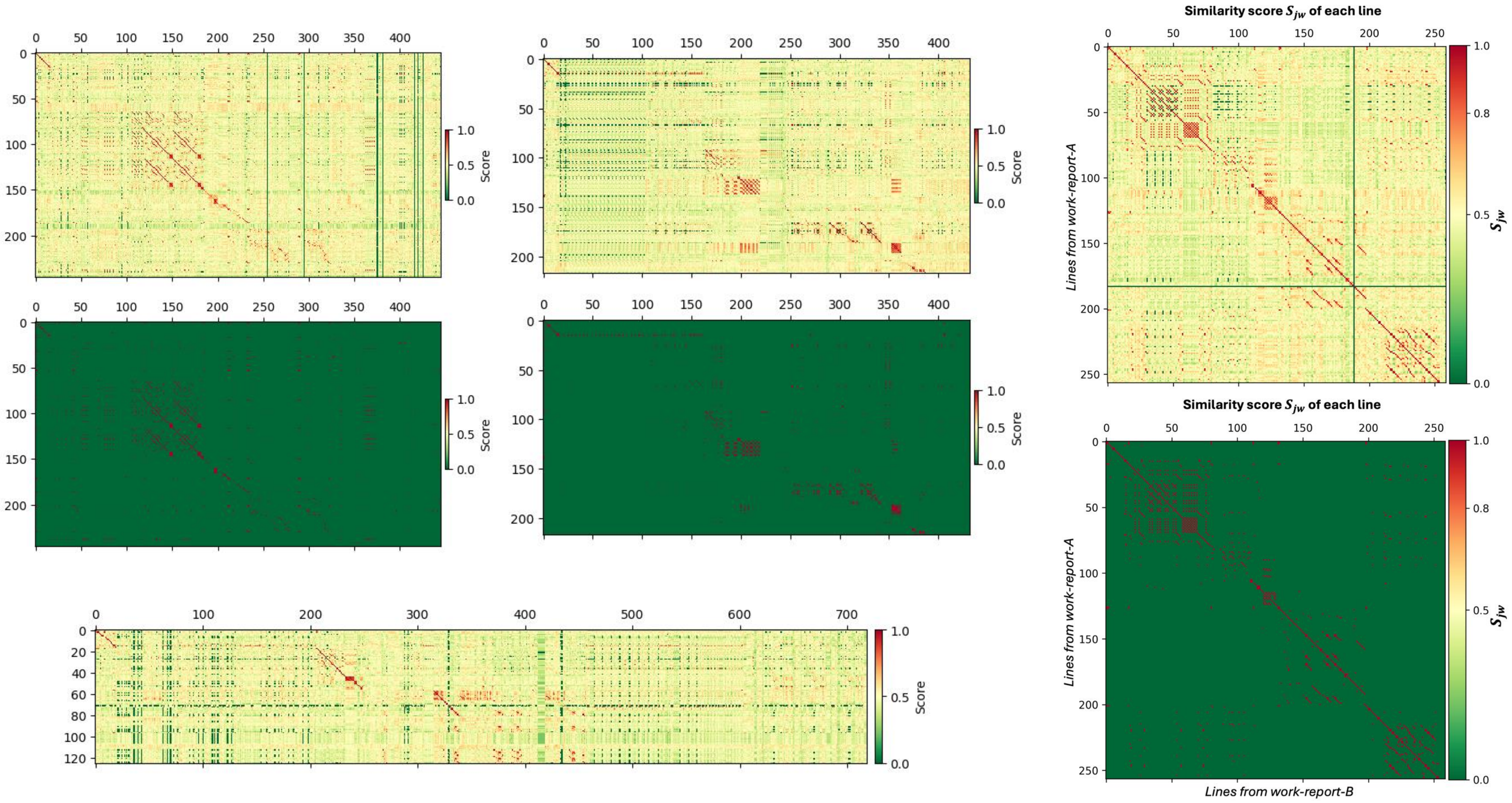
## 7 Conclusions

Your conclusions, suggestions for improvements, etc should go here.

## 8 References

Acknowledge others here (books, papers, repositories, libraries, tools)

# Example of Jaro–Winkler similarity



# Jaro–Winkler similarity

Jaro–Winkler similarity ***S*** is based on edit distance between two strings

$S \in [0,1]$

**Example**:

*$c_1$ = for i in range(10):*
*$c_2$ = for k in range(10):*
*$c_3$ = for k in LIST 10:*
*$c_4$ = for k in LIST A:*

$S_{c1,c2}$ = 0.9678
$S_{c1,c3}$ = 0.8270
$S_{c1,c4}$ = 0.6417

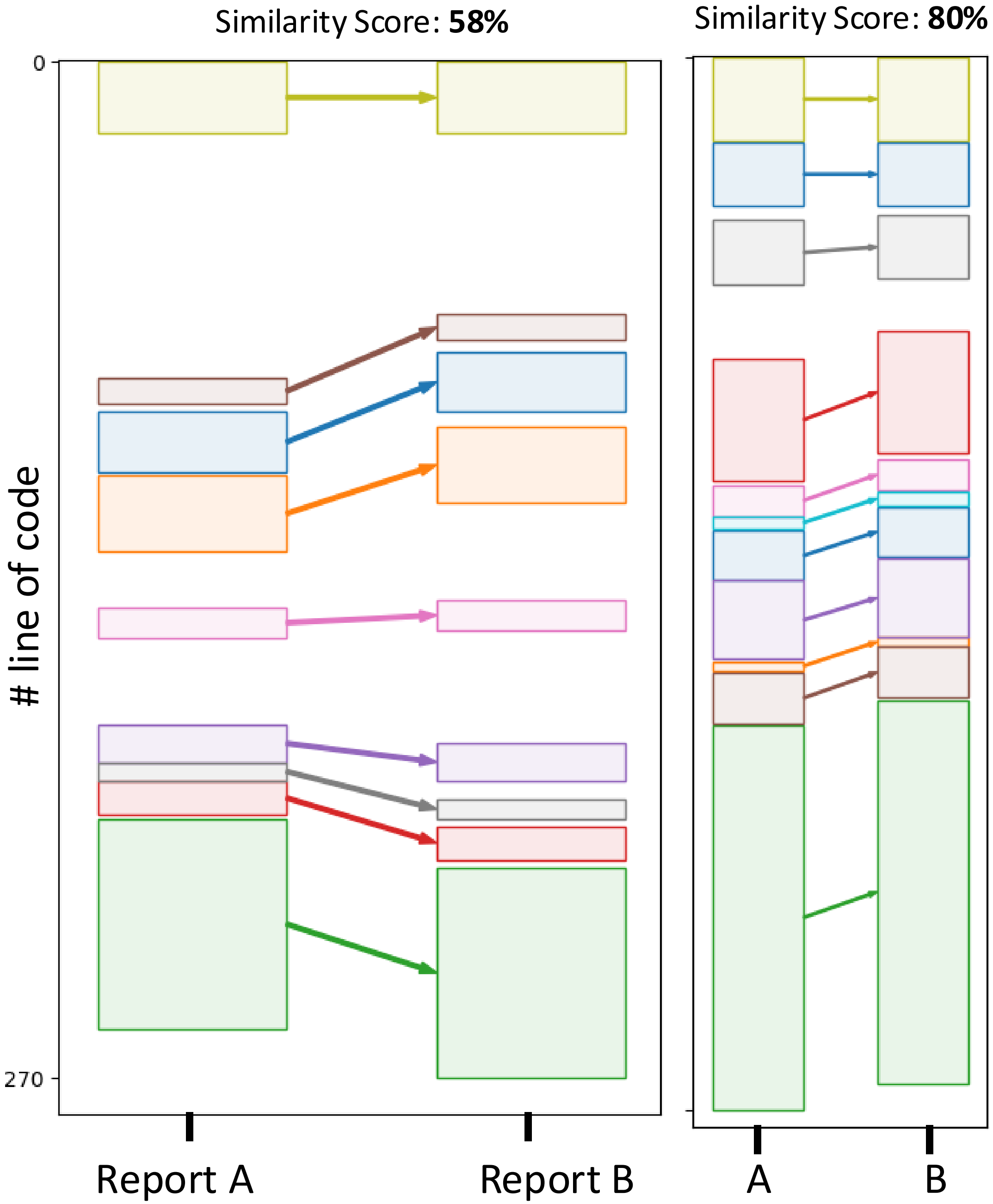
- Jaro similarity

$$sim_j = \begin{cases} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & else \end{cases}$$

- **Jaro-Winkler similarity**

$$sim_w = sim_j + lp(1 - sim_j)$$

Ref: https://en.wikipedia.org/wiki/Jaro%E2%80%93Winkler_distance


Similarity Score: **58%**
Similarity Score: **80%**
0
270
# line of code
Report A
Report B
A
B

Queen Mary

University of London