

## Introduction to Google Colab

**Instructor's Notes:** "Before we dive into the code, let's talk about Google Colab. Google Colab is a free cloud service that supports Python and allows you to write and execute Python code in your browser. It's especially useful for data science and machine learning projects. To get started, open your browser and go to [Google Colab](#). You'll need a Google account to use it."

### Creating a New Notebook in Google Colab

#### 1. Open Google Colab:

- Go to [Google Colab](#).
- Sign in with your Google account if you haven't already.

#### 2. Create a New Notebook:

- Click on "File" in the top-left corner.
- Select "New Notebook" from the dropdown menu.

#### 3. Rename Your Notebook:

- Click on the default title "Untitled" at the top and rename it to something like "SQuAD Model Assessment".

### Explanation of the Code

**Instructor's Notes:** "Now let's walk through the code step by step. This notebook demonstrates how to assess a Hugging Face question answering model using the Stanford Question Answering Dataset (SQuAD) and the [responsibleai](#) API."

### Markdown Cells

#### 1. Markdown Cells:

- The first few cells are in Markdown format. These cells are used to provide explanations and structure to the notebook.
- **Instructor's Notes:** "Markdown is a lightweight markup language that you can use to add formatting elements to plaintext text documents. In a Jupyter notebook, Markdown cells are used for documentation."

```
# Assess predictions on Stanford Question Answering Dataset (SQuAD) with a huggingface question answering model
```

- **Instructor's Notes:** "This is a Markdown cell with a header. The `#` symbol indicates a level 1 header."

```
This notebook demonstrates the use of the `responsibleai` API to assess a huggingface question answering model on the SQuAD dataset (see https://huggingface.co/datasets/squad for more information about the
```

dataset). It walks through the API calls necessary to create a widget with model analysis insights, then guides a visual analysis of the model.

- **Instructor's Notes:** "This cell provides an introduction and context for the notebook. It explains what we will be doing and provides a link to more information about the SQuAD dataset."

- [Launch Responsible AI Toolbox](#Launch-Responsible-AI-Toolbox)
- [Load Model and Data](#Load-Model-and-Data)
- [Create Model and Data Insights](#Create-Model-and-Data-Insights)

- **Instructor's Notes:** "This is a table of contents for the notebook, providing quick links to different sections."

## ## Launch Responsible AI Toolbox

- **Instructor's Notes:** "This is a level 2 header, introducing the section on launching the Responsible AI Toolbox."

The following section examines the code necessary to create datasets and a model. It then generates insights using the `responsibleai` API that can be visually analyzed.

- **Instructor's Notes:** "This cell provides an overview of what we will do in the next section."

### ### Prepare

To run this notebook, we need to install the following packages:

raiutils raiwidgets datasets transformers responsibleai\_text torch

Run the following command to load the spacy pipeline:

```
```bash
python -m spacy download en_core_web_sm
```

- **Instructor's Notes:** "This cell lists the packages we need to install and provides a command to load the Spacy pipeline. We'll install these packages using pip."

### ### Code Cells

#### 2. **\*\*Code Cells:\*\***

– The code cells contain Python code that will be executed to perform various tasks.

– **\*\*Instructor's Notes:\*\*** "The code cells contain the actual code we will run. Let's go through them one by one."

```
```python
import datasets
import pandas as pd
from transformers import pipeline
```

- **Instructor's Notes:** "Here, we're importing necessary libraries. `datasets` is used to load the SQuAD dataset, `pandas` is used for data manipulation, and `transformers` from Hugging Face is used to load the question-answering model."

```
dataset = datasets.load_dataset("squad", split="train")
dataset
```

- **Instructor's Notes:** "We load the SQuAD dataset using the `datasets` library. The `split='train'` parameter specifies that we're loading the training set."

```
questions = []
context = []
answers = []
for row in dataset:
    context.append(row['context'])
    questions.append(row['question'])
    answers.append(row['answers']['text'][0])
```

- **Instructor's Notes:** "We extract the context, questions, and answers from the dataset and store them in separate lists."

```
data = pd.DataFrame({'context': context, 'questions': questions,
                    'answers': answers})
data = data.sample(frac=1.0, random_state=42).reset_index(drop=True)
data.head()
```

- **Instructor's Notes:** "We create a pandas DataFrame from the lists and shuffle the data using `sample` with `frac=1.0` and `random_state=42` for reproducibility. `reset_index` is used to reset the index after shuffling."

```
pipeline_model = pipeline('question-answering')
test_size = 5

train_data = data
test_data = data[:test_size]
```

- **Instructor's Notes:** "We load a pre-trained question-answering model using the Hugging Face `pipeline` function. We also split the data into training and test sets, with the test set containing 5 samples."

```
def get_answer(dataset, idx):
    model_output = pipeline_model(question=dataset['questions'][idx],
                                   context=dataset['context'][idx])
    pred = model_output['answer']
    return pred

def check_answer(dataset, idx):
    pred = get_answer(dataset, idx)
    print('Question : ', dataset['questions'][idx])
    print('Answer : ', dataset['answers'][idx])
    print('Predicted : ', pred)
    print('Correct : ', pred == dataset['answers'][idx])

check_answer(test_data, 0)
```

- **Instructor's Notes:** "We define two functions: `get_answer` to get the model's prediction for a given question and context, and `check_answer` to compare the prediction with the actual answer. We then test the model on the first sample in the test set."

```
from responsibleai_text import RAITextInsights, ModelTask
from raiwidgets import ResponsibleAIDashboard
```

- **Instructor's Notes:** "We import the necessary classes from `responsibleai_text` and `raiwidgets` for model assessment and visualization."

```
rai_insights = RAITextInsights(pipeline_model, test_data, "answers",
                                task_type=ModelTask.QUESTION_ANSWERING)
```

- **Instructor's Notes:** "We initialize a `RAITextInsights` object with the model, test data, the target column (`answers`), and the task type (`ModelTask.QUESTION_ANSWERING`)."

```
rai_insights.error_analysis.add()
rai_insights.explainer.add()
```

- **Instructor's Notes:** "We add error analysis and explanation components to the `RAITextInsights` object."

```
rai_insights.compute()
```

- **Instructor's Notes:** "We compute the insights on the test set. This step processes the data and prepares it for visualization."

```
ResponsibleAIDashboard(rai_insights)
```

- **Instructor's Notes:** "Finally, we visualize and explore the model insights using the `ResponsibleAIDashboard`. This will display an interactive widget in the notebook."

## Running the Notebook in Google Colab

**Instructor's Notes:** "Now let's run this notebook in Google Colab. Follow these steps:"

### 1. Install Required Packages:

- In the first code cell, install the required packages by running:

```
!pip install raiutils raiwidgets datasets transformers  
responsibleai_text torch
```

- Install the Spacy pipeline by running:

```
!python -m spacy download en_core_web_sm
```

### 2. Copy and Paste Code:

- Copy and paste the provided code into separate code cells in your Colab notebook.
- Ensure each block of code is in its own cell to execute them step by step.

### 3. Run Each Code Cell:

- Click the "Run" button (a play icon) on the left side of each code cell to execute the code.
- Proceed sequentially, ensuring each cell runs without errors before moving to the next.

### 4. Visualize Results:

- After running all the cells, the `ResponsibleAIDashboard` cell will display an interactive dashboard in the notebook.
- You can explore the model insights and error analysis through this dashboard.

**Instructor's Notes:** "By following these steps, you can successfully run the notebook and analyze the question answering model on the SQuAD dataset using Google Colab. Feel free to ask any questions if you encounter issues or need further clarification."

---

This detailed explanation and step-by-step guide should help your students understand the code and how to use Google Colab effectively.