1 Prerequisites and experiment setup

The system this project is running on is on PopOS 22.04 LTS which is based on Ubuntu 22.04 LTS. In Project 1, I chose to investigate and optimize the learning rate, train batch size and optimizer type hyperparameters, the optimal values that were found were 6.991860954982597e-05, 256, Adam respectively.

2 Task 1: Migrate Notebook to Python Scripts

The Python Notebook was migrated into a single Python script. This script can receive arguments upon running the script like so:

```
python distilbert_on_mrpc.py --learning_rate 1e-05
```

The arguments are parsed with 'argparse' and added to the model. If a specific or no arguments were passed, then it uses the default values.

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument('--learning_rate', dest='learning_rate',
    type=float)
parser.add_argument('--adam_epsilon', dest='adam_epsilon',
    type=float)
...
```

The experiments are logged using wandb.ai.

3 Task 2: Creating a Docker Image

The created docker image contains the python script and installs all required libraries. When running the container, just like in task 1, the hyperparameters can be passed as arguments. The following command runs a single training run with the best parameters:

```
    sudo docker run run_distilbert_training python3 ./distilbert_on_mrpc.py
--api_key YOUR_WANDB_API_KEY --learning_rate 6.991860954982597e-05
--train_batch_size 256 --optimizer_type Adam
```

NOTE: The wandb api key is a parameter here. I am aware that this does not follow standards. In a company setting, the key would be in a safe place and retrieved from there. But just for the sake of this school project, it makes it easier to test for other students.

This is how the docker image is installed and structured:

```
# syntax=docker/dockerfile:1
FROM python:3.10-slim
WORKDIR /usr/src/app
COPY ./distilbert_on_mrpc.py ./
RUN pip install scipy
RUN pip install scikit-learn
RUN pip install wandb
RUN pip install torch --index-url
    https://download.pytorch.org/whl/cpu
```

```
RUN pip install transformers
RUN pip install pytorch_lightning==1.9.5
RUN pip install datasets
CMD ["python3", "./distilbert_on_mrpc.py"]
```

... by running the command: `sudo docker build -t run_distilbert_training .`

4 Task 3: Running the Docker Image on Playground

Setup: On docker playground, I cloned my GitHub repository that contains the code and the dockerfile (this repository is the same as in task 4). I then build the image and ran a docker container.

I wanted to use the same parameters as in task 2, however I noticed that the playground was very resource limited. I went from 256 batch size to 16 (with the rest of the parameter being default). Computation wise took much longer on the playground compared to my laptop which only had a cpu. A single epoch to train took more than one hour. I also had to change the torch libary to the only-cpu torch library. The playground did not have enough space to download regular torch. Below are the accuracy and F1 scores of the local run and the Playground run. Both use the hyperparameter values. Model performance wise stayed the same:

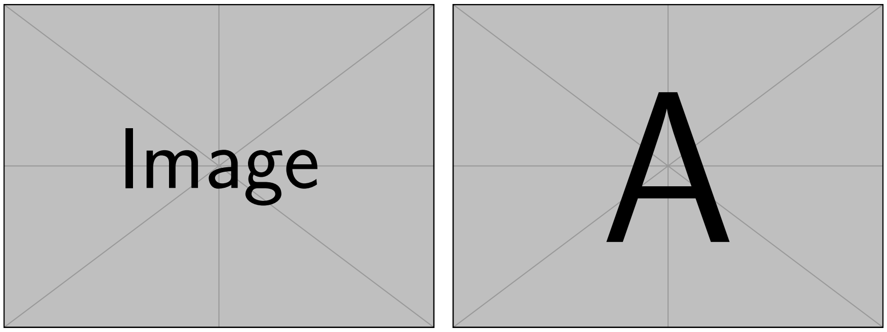

Figure 1: Accuracy and F1 Scores on Local vs Playground

5 Task 4: Setup GitHub Repository

Here is a link to my private project. You can find additional information and help under the readme section. GitHub Project 2.

6 Reflection

It took for me a lot of time to setup the docker image file for it to be able to take in hyperparameter arguments. I have never worked with docker before, so I expected this. I also had a hard time with using Docker Playground. I have made many attempts on trying to build the Docker image file it somehow always gave a no space error. It was quite frustrating. Task 1 and Task 4 were done fairly quickly, since I have already worked with GitHub in the past.