Notebook

April 20, 2025

[1]:
```
from google.colab import drive
drive.mount('/content/drive/')
```

```
Mounted at /content/drive/
```

[2]:
```
!pip install mlflow scikit-learn
!pip install torch torchvision
!pip install transformers datasets
```

```
Collecting mlflow
  Downloading mlflow-2.21.3-py3-none-any.whl.metadata (30 kB)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-
packages (1.6.1)
Collecting mlflow-skinny==2.21.3 (from mlflow)
  Downloading mlflow_skinny-2.21.3-py3-none-any.whl.metadata (31 kB)
Requirement already satisfied: Flask<4 in /usr/local/lib/python3.11/dist-
packages (from mlflow) (3.1.0)
Requirement already satisfied: Jinja2<4,>=2.11 in
/usr/local/lib/python3.11/dist-packages (from mlflow) (3.1.6)
Collecting alembic!=1.10.0,<2 (from mlflow)
  Downloading alembic-1.15.2-py3-none-any.whl.metadata (7.3 kB)
Collecting docker<8,>=4.0.0 (from mlflow)
  Downloading docker-7.1.0-py3-none-any.whl.metadata (3.8 kB)
Collecting graphene<4 (from mlflow)
  Downloading graphene-3.4.3-py2.py3-none-any.whl.metadata (6.9 kB)
Collecting gunicorn<24 (from mlflow)
  Downloading gunicorn-23.0.0-py3-none-any.whl.metadata (4.4 kB)
Requirement already satisfied: markdown<4,>=3.3 in
/usr/local/lib/python3.11/dist-packages (from mlflow) (3.8)
Requirement already satisfied: matplotlib<4 in /usr/local/lib/python3.11/dist-
packages (from mlflow) (3.10.0)
Requirement already satisfied: numpy<3 in /usr/local/lib/python3.11/dist-
packages (from mlflow) (2.0.2)
Requirement already satisfied: pandas<3 in /usr/local/lib/python3.11/dist-
packages (from mlflow) (2.2.2)
Requirement already satisfied: pyarrow<20,>=4.0.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow) (18.1.0)
Requirement already satisfied: scipy<2 in /usr/local/lib/python3.11/dist-
packages (from mlflow) (1.14.1)
```

```
Requirement already satisfied: sqlalchemy<3,>=1.4.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow) (2.0.40)
Requirement already satisfied: cachetools<6,>=5.0.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(5.5.2)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.11/dist-
packages (from mlflow-skinny==2.21.3->mlflow) (8.1.8)
Requirement already satisfied: cloudpickle<4 in /usr/local/lib/python3.11/dist-
packages (from mlflow-skinny==2.21.3->mlflow) (3.1.1)
Collecting databricks-sdk<1,>=0.20.0 (from mlflow-skinny==2.21.3->mlflow)
  Downloading databricks_sdk-0.50.0-py3-none-any.whl.metadata (38 kB)
Collecting fastapi<1 (from mlflow-skinny==2.21.3->mlflow)
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Requirement already satisfied: gitpython<4,>=3.1.9 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(3.1.44)
Requirement already satisfied: importlib_metadata!=4.7.0,<9,>=3.7.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(8.6.1)
Requirement already satisfied: opentelemetry-api<3,>=1.9.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(1.32.1)
Requirement already satisfied: opentelemetry-sdk<3,>=1.9.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(1.32.1)
Requirement already satisfied: packaging<25 in /usr/local/lib/python3.11/dist-
packages (from mlflow-skinny==2.21.3->mlflow) (24.2)
Requirement already satisfied: protobuf<6,>=3.12.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(5.29.4)
Requirement already satisfied: pydantic<3,>=1.10.8 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(2.11.3)
Requirement already satisfied: pyyaml<7,>=5.1 in /usr/local/lib/python3.11/dist-
packages (from mlflow-skinny==2.21.3->mlflow) (6.0.2)
Requirement already satisfied: requests<3,>=2.17.3 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(2.32.3)
Requirement already satisfied: sqlparse<1,>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(0.5.3)
Requirement already satisfied: typing-extensions<5,>=4.0.0 in
/usr/local/lib/python3.11/dist-packages (from mlflow-skinny==2.21.3->mlflow)
(4.13.2)
Collecting uvicorn<1 (from mlflow-skinny==2.21.3->mlflow)
  Downloading uvicorn-0.34.2-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-
packages (from scikit-learn) (1.4.2)
```

```
Requirement already satisfied: threadpoolctl>=3.1.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: Mako in /usr/lib/python3/dist-packages (from
alembic!=1.10.0,<2->mlflow) (1.1.3)
Requirement already satisfied: urllib3>=1.26.0 in
/usr/local/lib/python3.11/dist-packages (from docker<8,>=4.0.0->mlflow) (2.3.0)
Requirement already satisfied: Werkzeug>=3.1 in /usr/local/lib/python3.11/dist-
packages (from Flask<4->mlflow) (3.1.3)
Requirement already satisfied: itsdangerous>=2.2 in
/usr/local/lib/python3.11/dist-packages (from Flask<4->mlflow) (2.2.0)
Requirement already satisfied: blinker>=1.9 in /usr/local/lib/python3.11/dist-
packages (from Flask<4->mlflow) (1.9.0)
Collecting graphql-core<3.3,>=3.1 (from graphene<4->mlflow)
  Downloading graphql_core-3.2.6-py3-none-any.whl.metadata (11 kB)
Collecting graphql-relay<3.3,>=3.1 (from graphene<4->mlflow)
  Downloading graphql_relay-3.2.0-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: python-dateutil<3,>=2.7.0 in
/usr/local/lib/python3.11/dist-packages (from graphene<4->mlflow) (2.8.2)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/dist-packages (from Jinja2<4,>=2.11->mlflow) (3.0.2)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<4->mlflow) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-
packages (from matplotlib<4->mlflow) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<4->mlflow) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<4->mlflow) (1.4.8)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-
packages (from matplotlib<4->mlflow) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<4->mlflow) (3.2.3)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
packages (from pandas<3->mlflow) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-
packages (from pandas<3->mlflow) (2025.2)
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.11/dist-
packages (from sqlalchemy<3,>=1.4.0->mlflow) (3.2.0)
Requirement already satisfied: google-auth~=2.0 in
/usr/local/lib/python3.11/dist-packages (from databricks-sdk<1,>=0.20.0->mlflow-
skinny==2.21.3->mlflow) (2.38.0)
Collecting starlette<0.47.0,>=0.40.0 (from fastapi<1->mlflow-
skinny==2.21.3->mlflow)
  Downloading starlette-0.46.2-py3-none-any.whl.metadata (6.2 kB)
Requirement already satisfied: gitdb<5,>=4.0.1 in
/usr/local/lib/python3.11/dist-packages (from gitpython<4,>=3.1.9->mlflow-
skinny==2.21.3->mlflow) (4.0.12)
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.11/dist-
```

```
packages (from importlib_metadata!=4.7.0,<9,>=3.7.0->mlflow-
skinny==2.21.3->mlflow) (3.21.0)
Requirement already satisfied: deprecated>=1.2.6 in
/usr/local/lib/python3.11/dist-packages (from opentelemetry-
api<3,>=1.9.0->mlflow-skinny==2.21.3->mlflow) (1.2.18)
Requirement already satisfied: opentelemetry-semantic-conventions==0.53b1 in
/usr/local/lib/python3.11/dist-packages (from opentelemetry-
sdk<3,>=1.9.0->mlflow-skinny==2.21.3->mlflow) (0.53b1)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.10.8->mlflow-
skinny==2.21.3->mlflow) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.1 in
/usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.10.8->mlflow-
skinny==2.21.3->mlflow) (2.33.1)
Requirement already satisfied: typing-inspection>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.10.8->mlflow-
skinny==2.21.3->mlflow) (0.4.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil<3,>=2.7.0->graphene<4->mlflow) (1.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.17.3->mlflow-
skinny==2.21.3->mlflow) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
packages (from requests<3,>=2.17.3->mlflow-skinny==2.21.3->mlflow) (3.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.17.3->mlflow-
skinny==2.21.3->mlflow) (2025.1.31)
Requirement already satisfied: h11>=0.8 in /usr/local/lib/python3.11/dist-
packages (from uvicorn<1->mlflow-skinny==2.21.3->mlflow) (0.14.0)
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.11/dist-
packages (from deprecated>=1.2.6->opentelemetry-api<3,>=1.9.0->mlflow-
skinny==2.21.3->mlflow) (1.17.2)
Requirement already satisfied: smmap<6,>=3.0.1 in
/usr/local/lib/python3.11/dist-packages (from
gitdb<5,>=4.0.1->gitpython<4,>=3.1.9->mlflow-skinny==2.21.3->mlflow) (5.0.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.11/dist-packages (from google-auth~=2.0->databricks-
sdk<1,>=0.20.0->mlflow-skinny==2.21.3->mlflow) (0.4.2)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.11/dist-
packages (from google-auth~=2.0->databricks-sdk<1,>=0.20.0->mlflow-
skinny==2.21.3->mlflow) (4.9.1)
Requirement already satisfied: anyio<5,>=3.6.2 in
/usr/local/lib/python3.11/dist-packages (from
starlette<0.47.0,>=0.40.0->fastapi<1->mlflow-skinny==2.21.3->mlflow) (4.9.0)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-
packages (from anyio<5,>=3.6.2->starlette<0.47.0,>=0.40.0->fastapi<1->mlflow-
skinny==2.21.3->mlflow) (1.3.1)
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in
```

```
/usr/local/lib/python3.11/dist-packages (from pyasn1-modules>=0.2.1->google-
auth~=2.0->databricks-sdk<1,>=0.20.0->mlflow-skinny==2.21.3->mlflow) (0.6.1)
Downloading mlflow-2.21.3-py3-none-any.whl (28.2 MB)
                         28.2/28.2 MB
65.4 MB/s eta 0:00:00
Downloading mlflow_skinny-2.21.3-py3-none-any.whl (6.1 MB)
                         6.1/6.1 MB
108.4 MB/s eta 0:00:00
Downloading alembic-1.15.2-py3-none-any.whl (231 kB)
                         231.9/231.9 kB
20.6 MB/s eta 0:00:00
Downloading docker-7.1.0-py3-none-any.whl (147 kB)
                         147.8/147.8 kB
14.1 MB/s eta 0:00:00
Downloading graphene-3.4.3-py2.py3-none-any.whl (114 kB)
                         114.9/114.9 kB
11.1 MB/s eta 0:00:00
Downloading gunicorn-23.0.0-py3-none-any.whl (85 kB)
                         85.0/85.0 kB
8.0 MB/s eta 0:00:00
Downloading databricks_sdk-0.50.0-py3-none-any.whl (692 kB)
                         692.3/692.3 kB
46.1 MB/s eta 0:00:00
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
                         95.2/95.2 kB
8.9 MB/s eta 0:00:00
Downloading graphql_core-3.2.6-py3-none-any.whl (203 kB)
                         203.4/203.4 kB
16.8 MB/s eta 0:00:00
Downloading graphql_relay-3.2.0-py3-none-any.whl (16 kB)
Downloading uvicorn-0.34.2-py3-none-any.whl (62 kB)
                         62.5/62.5 kB
5.5 MB/s eta 0:00:00
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
                         72.0/72.0 kB
6.5 MB/s eta 0:00:00
Installing collected packages: uvicorn, gunicorn, graphql-core, starlette,
graphql-relay, docker, alembic, graphene, fastapi, databricks-sdk, mlflow-
skinny, mlflow
Successfully installed alembic-1.15.2 databricks-sdk-0.50.0 docker-7.1.0
fastapi-0.115.12 graphene-3.4.3 graphql-core-3.2.6 graphql-relay-3.2.0
gunicorn-23.0.0 mlflow-2.21.3 mlflow-skinny-2.21.3 starlette-0.46.2
uvicorn-0.34.2
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-packages
(2.6.0+cu124)
Requirement already satisfied: torchvision in /usr/local/lib/python3.11/dist-
packages (0.21.0+cu124)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-
```

```
packages (from torch) (3.18.0)
Requirement already satisfied: typing-extensions>=4.10.0 in
/usr/local/lib/python3.11/dist-packages (from torch) (4.13.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-
packages (from torch) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages
(from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages
(from torch) (2025.3.2)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu12==10.3.5.147 (from torch)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in
/usr/local/lib/python3.11/dist-packages (from torch) (0.6.2)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in
/usr/local/lib/python3.11/dist-packages (from torch) (2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in
/usr/local/lib/python3.11/dist-packages (from torch) (12.4.127)
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-
packages (from torch) (3.2.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-
```

```
packages (from torch) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.11/dist-packages (from sympy==1.13.1->torch) (1.3.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages
(from torchvision) (2.0.2)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in
/usr/local/lib/python3.11/dist-packages (from torchvision) (11.1.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/dist-packages (from jinja2->torch) (3.0.2)
Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-manylinux2014_x86_64.whl (363.4
MB)
                         363.4/363.4 MB
2.5 MB/s eta 0:00:00
Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (13.8 MB)
                         13.8/13.8 MB
109.6 MB/s eta 0:00:00
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (24.6 MB)
                         24.6/24.6 MB
75.0 MB/s eta 0:00:00
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (883 kB)
                         883.7/883.7 kB
48.7 MB/s eta 0:00:00
Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-manylinux2014_x86_64.whl
(664.8 MB)
                         664.8/664.8 MB
1.3 MB/s eta 0:00:00
Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-manylinux2014_x86_64.whl
(211.5 MB)
                         211.5/211.5 MB
5.4 MB/s eta 0:00:00
Downloading nvidia_curand_cu12-10.3.5.147-py3-none-
manylinux2014_x86_64.whl (56.3 MB)
                         56.3/56.3 MB
11.7 MB/s eta 0:00:00
Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-
manylinux2014_x86_64.whl (127.9 MB)
                         127.9/127.9 MB
8.0 MB/s eta 0:00:00
Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-
manylinux2014_x86_64.whl (207.5 MB)
                         207.5/207.5 MB
6.1 MB/s eta 0:00:00
Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl (21.1 MB)
                         21.1/21.1 MB
```

```
91.6 MB/s eta 0:00:00
Installing collected packages: nvidia-nvjitlink-cu12, nvidia-curand-cu12,
nvidia-cufft-cu12, nvidia-cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-
cuda-cupti-cu12, nvidia-cublas-cu12, nvidia-cusparse-cu12, nvidia-cudnn-cu12,
nvidia-cusolver-cu12
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-
cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-cu12-12.4.127
nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3 nvidia-curand-
```

```
cu12-10.3.5.147 nvidia-cusolver-cu12-11.6.1.9 nvidia-cusparse-cu12-12.3.1.170
nvidia-nvjitlink-cu12-12.4.127
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-
packages (4.51.3)
Collecting datasets
  Downloading datasets-3.5.0-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-
packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.30.2)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-
packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-
packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-
packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in
/usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-
packages (from transformers) (4.67.1)
Requirement already satisfied: pyarrow>=15.0.0 in
/usr/local/lib/python3.11/dist-packages (from datasets) (18.1.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages
(from datasets) (2.2.2)
Collecting xxhash (from datasets)
  Downloading
xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py311-none-any.whl.metadata (7.2 kB)
Collecting fsspec<=2024.12.0,>=2023.1.0 (from
fsspec[http]<=2024.12.0,>=2023.1.0->datasets)
  Downloading fsspec-2024.12.0-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-
packages (from datasets) (3.11.15)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-
```

```
packages (from aiohttp->datasets) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (6.4.3)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.19.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub<1.0,>=0.30.0->transformers) (4.13.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-
packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests->transformers)
(2025.1.31)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-
packages (from pandas->datasets) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-
packages (from pandas->datasets) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-
packages (from python-dateutil>=2.8.2->pandas->datasets) (1.17.0)
Downloading datasets-3.5.0-py3-none-any.whl (491 kB)
                         491.2/491.2 kB
8.7 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
                         116.3/116.3 kB
6.9 MB/s eta 0:00:00
Downloading fsspec-2024.12.0-py3-none-any.whl (183 kB)
                         183.9/183.9 kB
15.3 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py311-none-any.whl (143 kB)
                         143.5/143.5 kB
12.2 MB/s eta 0:00:00
Downloading
xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
                         194.8/194.8 kB
14.7 MB/s eta 0:00:00
Installing collected packages: xxhash, fsspec, dill, multiprocess,
datasets
  Attempting uninstall: fsspec
```

```
    Found existing installation: fsspec 2025.3.2
    Uninstalling fsspec-2025.3.2:
      Successfully uninstalled fsspec-2025.3.2
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
gcsfs 2025.3.2 requires fsspec==2025.3.2, but you have fsspec 2024.12.0 which is
incompatible.
Successfully installed datasets-3.5.0 dill-0.3.8 fsspec-2024.12.0
multiprocess-0.70.16 xxhash-3.5.0
```

[3]:
```
!ls 'drive/MyDrive/AML/Assignment 5/data/'
```

```
chicken-data  sentiment-data
```

0.1 Imports and global variables

[1]:
```
import os
import random
import mlflow
import time
from collections import Counter
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from PIL import Image

import torch
from torch.utils.data import DataLoader, WeightedRandomSampler, TensorDataset,␣
 ↪DataLoader, RandomSampler, SequentialSampler
from torch import nn
from torch.optim import Adam
from torch.nn import Linear, ReLU, Dropout, LayerNorm, Sequential,␣
 ↪CrossEntropyLoss
from torch.cuda import is_available
from torch.optim.lr_scheduler import ReduceLROnPlateau
from torch.nn.functional import softmax

from torchvision import models
from torchvision import transforms
from torchvision.models import ResNet50_Weights, MobileNet_V2_Weights,␣
 ↪EfficientNet_B0_Weights
from torchvision.datasets import ImageFolder
```

```
import transformers
from transformers import AutoModel, BertTokenizerFast

from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.utils.class_weight import compute_class_weight

device = "cuda" if is_available() else "cpu"

# a5_data_path = 'drive/MyDrive/AML/Assignment 5/data'
a5_data_path = './data'
```

```
C:\Users\Shankar\.conda\envs\DL-1\lib\site-packages\tqdm\auto.py:21:
TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

1 Part 1: Chicken or Duck

1.1 EDA and Preprocessing

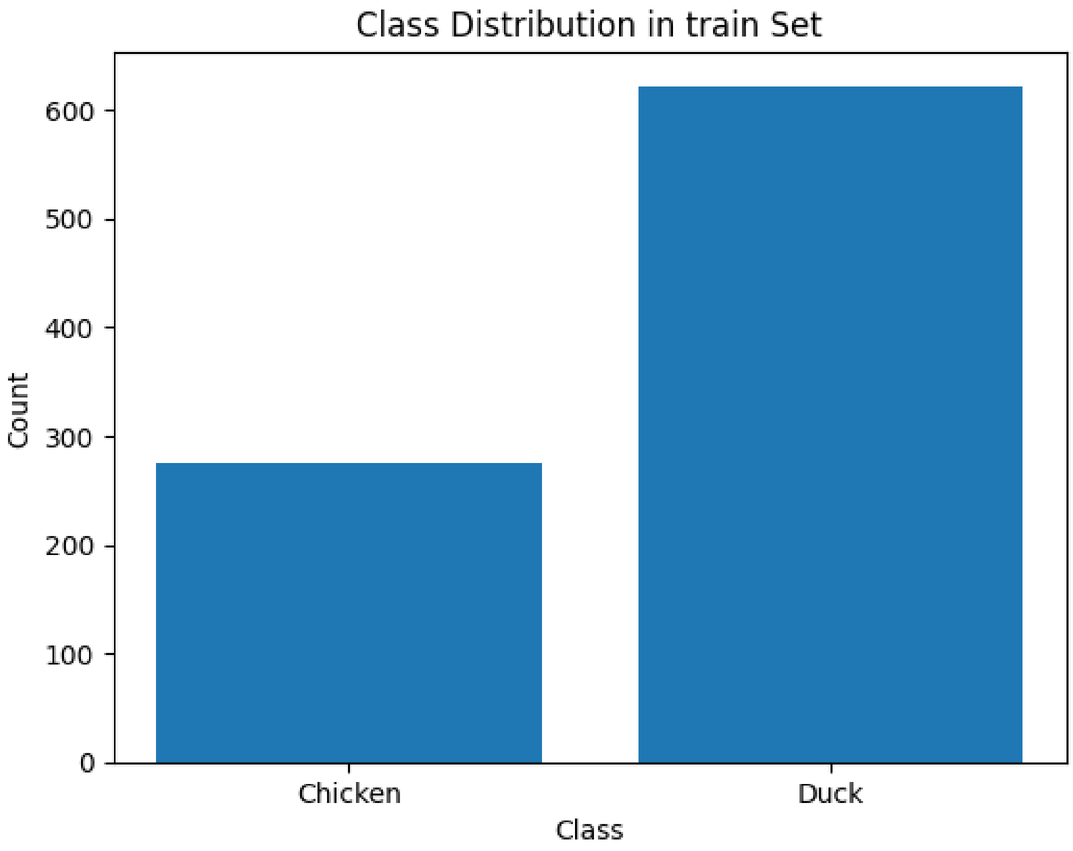
[5]:

```
folder_path = a5_data_path + '/chicken-data'

def files_in_directory(directory_path):
    return [file for file in os.listdir(directory_path) if os.path.isfile(os.
  ↪path.join(directory_path, file))]

def get_random_files(directory_path, num_files=5):
    all_files = [file for file in os.listdir(directory_path) if os.path.
  ↪isfile(os.path.join(directory_path, file))]
    random_files = random.sample(all_files, num_files) if len(all_files) >=␣
  ↪num_files else all_files
    return random_files

def load_image(file_path):
    img = Image.open(file_path).convert("RGB")
    return img

def dataset_summary(dataset):
    class_counts = {
        "Chicken": len(files_in_directory(folder_path + '/' + dataset + '/
  ↪chicken')),
        "Duck": len(files_in_directory(folder_path + '/' + dataset + '/duck'))
    }
    print(f"{dataset} Set Summary:")
    print(f"Total Samples: {len(dataset)}")
```

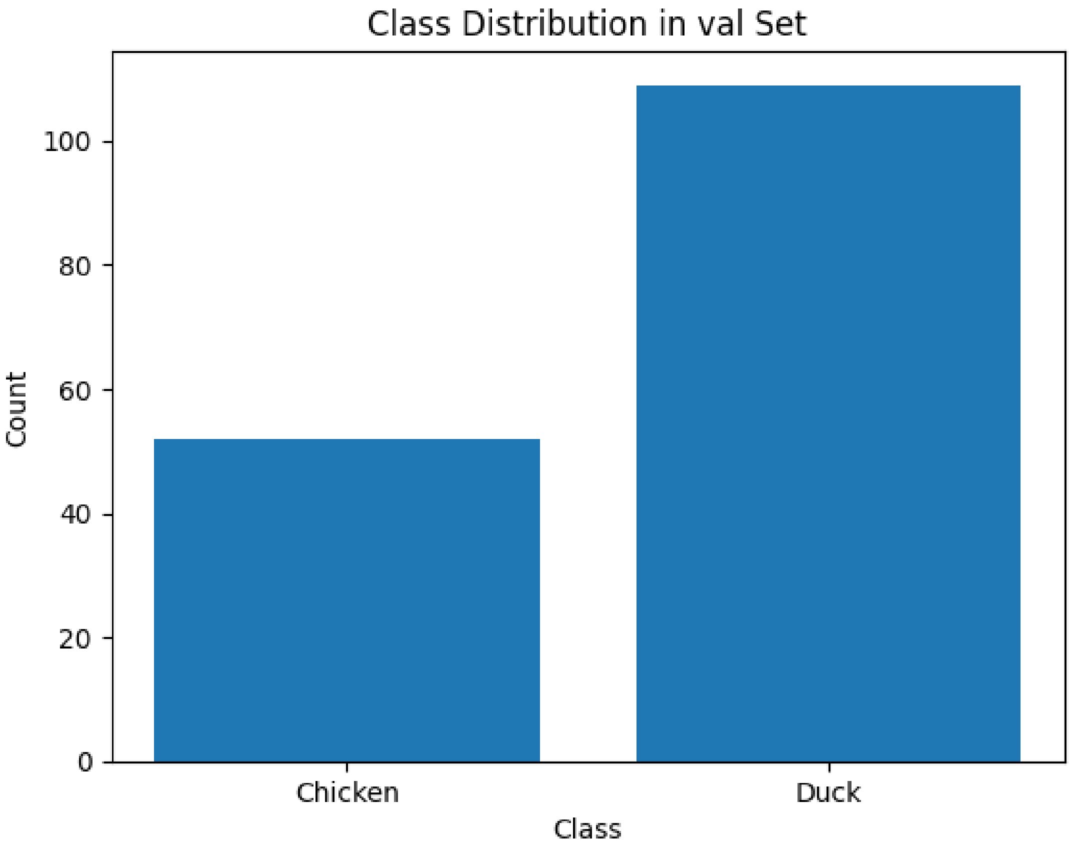
```python
    print(f"Class Distribution: {dict(class_counts)}")
    print()
    plt.bar(class_counts.keys(), class_counts.values(), tick_label=["Chicken", "Duck"])
    plt.title(f"Class Distribution in {dataset} Set")
    plt.xlabel("Class")
    plt.ylabel("Count")
    plt.show()

def show_random_images(dataset, n=5):
    plt.figure(figsize=(15, 5))
    random_chickens = get_random_files(folder_path + '/' + dataset + '/chicken',n)
    random_ducks = get_random_files(folder_path + '/' + dataset + '/duck',n)

    for i in range(n):
        img = load_image(folder_path + '/' + dataset + '/chicken/' + random_chickens[i])
        plt.subplot(2, n, i+1)
        plt.imshow(img)
        plt.title(f"Label: Chicken")
        plt.axis("off")

    for i in range(n):
        img = load_image(folder_path + '/' + dataset + '/duck/' + random_ducks[i])
        plt.subplot(2, n, i+6)
        plt.imshow(img)
        plt.title(f"Label: Duck")
        plt.axis("off")
    plt.show()

def inspect_image_data(dataset):
    heights = []
    widths = []
    pixel_means = []
    pixel_stds = []

    fpth = folder_path + '/' + dataset + '/chicken'
    for file in files_in_directory(fpth):
        img = np.array(load_image(fpth + '/' + file))
        heights.append(img.shape[1])
        widths.append(img.shape[2])
        pixel_means.append(np.mean(img))
        pixel_stds.append(np.std(img))

    fpth = folder_path + '/' + dataset + '/duck'
    for file in files_in_directory(fpth):
        img = np.array(load_image(fpth + '/' + file))
        heights.append(img.shape[1])
        widths.append(img.shape[2])
        pixel_means.append(np.mean(img))
        pixel_stds.append(np.std(img))

    print(f"Image Height: Min = {min(heights)}, Max = {max(heights)}")
    print(f"Image Width: Min = {min(widths)}, Max = {max(widths)}")
    print(f"Pixel Intensity Mean: {np.mean(pixel_means):.4f}")
    print(f"Pixel Intensity Std: {np.mean(pixel_stds):.4f}")

# Define transformation
my_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomHorizontalFlip(),
    transforms.RandomRotation(30),
    transforms.ColorJitter(),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])␣
  ↪#ImageNet Weights
])


def denormalize(normalized_tensor):
    mean = torch.tensor([0.485, 0.456, 0.406]).view(3, 1, 1)
    std = torch.tensor([0.229, 0.224, 0.225]).view(3, 1, 1)
    denormalized_tensor = normalized_tensor * std + mean  # Reverse␣
  ↪normalization
    return denormalized_tensor

def show_transformed_images(dataset, n=5):
    plt.figure(figsize=(15, 5))
    random_chickens = get_random_files(folder_path + '/' + dataset + '/
  ↪chicken',n)
    random_ducks = get_random_files(folder_path + '/' + dataset + '/duck',n)

    for i in range(n):
        img = load_image(folder_path + '/' + dataset + '/chicken/' +␣
  ↪random_chickens[i])
        transformed_img = denormalize(my_transform(img))
        transformed_img = transforms.ToPILImage()(transformed_img)
        plt.subplot(2, n, i+1)
        plt.imshow(transformed_img)
        plt.title(f"Label: Transformed Chicken")
        plt.axis("off")
    for i in range(n):
        img = load_image(folder_path + '/' + dataset + '/duck/' +␣
↪random_ducks[i])
        transformed_img = denormalize(my_transform(img))
        transformed_img = transforms.ToPILImage()(transformed_img)

        plt.subplot(2, n, i+6)
        plt.imshow(transformed_img)
        plt.title(f"Label: Transformed Duck")
        plt.axis("off")
    plt.show()
```

[6]:
```
# Print summaries for all datasets
dataset_summary("train")
dataset_summary("val")
dataset_summary("test")
```

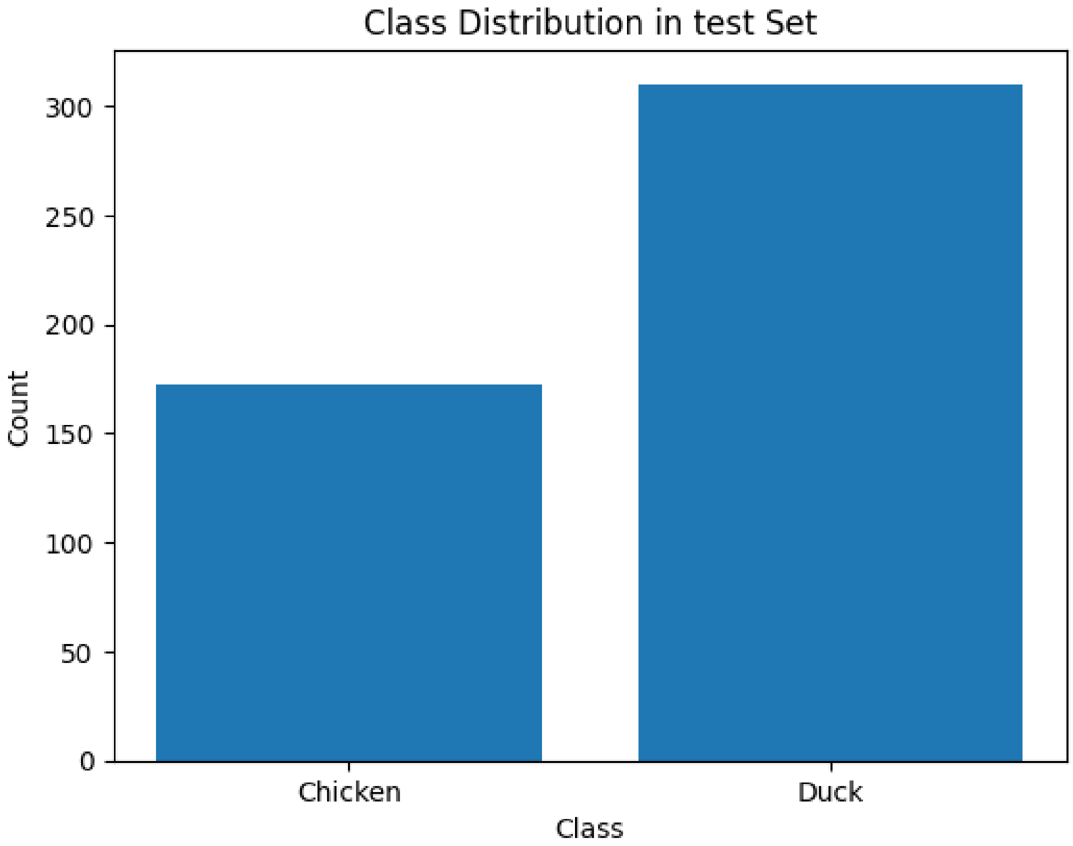
```
train Set Summary:
Total Samples: 5
Class Distribution: {'Chicken': 275, 'Duck': 622}
```



```
val Set Summary:
Total Samples: 3
Class Distribution: {'Chicken': 52, 'Duck': 109}
```

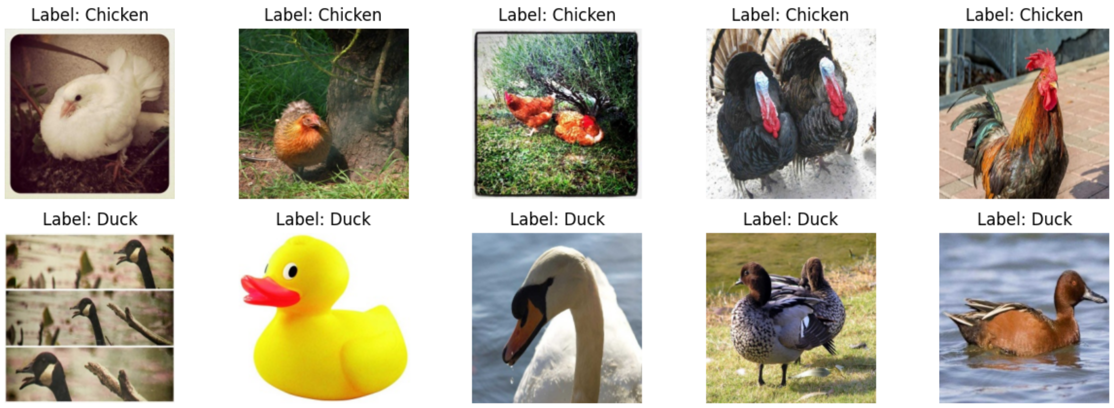


```
test Set Summary:
Total Samples: 4
Class Distribution: {'Chicken': 172, 'Duck': 310}
```



[7]:

```
# Visualize random images
show_random_images('train')
```


Label: Chicken Label: Chicken Label: Chicken Label: Chicken Label: Chicken

Label: Duck Label: Duck Label: Duck Label: Duck Label: Duck

[8]:

```
# Inspect train dataset
inspect_image_data('train')
```

```
Image Height: Min = 256, Max = 256
Image Width: Min = 3, Max = 3
Pixel Intensity Mean: 124.2285
Pixel Intensity Std: 54.9441
```

[9]:

```python
# Apply and show Transformation
show_transformed_images('train')
```

Label: Transformed Chicken
Label: Transformed Chicken
Label: Transformed Chicken
Label: Transformed Chicken
Label: Transformed Chicken
Label: Transformed Duck
Label: Transformed Duck
Label: Transformed Duck
Label: Transformed Duck
Label: Transformed Duck

1.2 Training

[53]:

```python
# Weighted Random Sampler
def get_sampler(dataset):
    labels = [label for _, label in dataset]
    class_sample_counts = Counter(labels)
    class_weights = 1. / torch.tensor([class_sample_counts[label] for label in␣
 ↪labels], dtype=torch.float)
    sample_weights = [class_weights[label] for label in labels]
    sampler = WeightedRandomSampler(sample_weights,␣
 ↪num_samples=len(sample_weights), replacement=True)
    return sampler

#Load Datasets
train_dataset = ImageFolder(folder_path + '/train', transform=my_transform)
val_dataset = ImageFolder(folder_path + '/val', transform=my_transform)
test_dataset = ImageFolder(folder_path + '/test', transform=my_transform)

train_sampler = get_sampler(train_dataset)
val_sampler = get_sampler(val_dataset)

# Create DataLoaders
train_loader = DataLoader(train_dataset, batch_size=16, sampler=train_sampler)
val_loader = DataLoader(val_dataset, batch_size=16, sampler=val_sampler)
test_loader = DataLoader(test_dataset, batch_size=16, shuffle=False)
```

[54]:

```
# Load pre-trained models (e.g., ResNet, MobileNet, EfficientNet)
def replace_classifier(in_features, num_classes=2):
    return Sequential(
        Linear(in_features, 8),
        ReLU(),
        Dropout(0.5),
        Linear(8, num_classes)
    )

def initialize_weights(module):
    if isinstance(module, Linear):
        nn.init.xavier_uniform_(module.weight)

def get_model(model_name, freeze_layers=False):
    if model_name == "resnet50":
        weights = ResNet50_Weights.DEFAULT
        model = models.resnet50(weights=weights)
        if freeze_layers:
            for param in model.parameters():
                param.requires_grad = False
        model.fc = replace_classifier(model.fc.in_features)
        model.fc.apply(initialize_weights)

    elif model_name == "mobilenet_v2":
        weights = MobileNet_V2_Weights.DEFAULT
        model = models.mobilenet_v2(weights=weights)
        if freeze_layers:
            for param in model.features.parameters():
                param.requires_grad = False
        model.classifier[-1] = replace_classifier(model.last_channel)
        model.classifier[-1].apply(initialize_weights)

    elif model_name == "efficientnet_b0":
        weights = EfficientNet_B0_Weights.DEFAULT
        model = models.efficientnet_b0(weights=weights)
        if freeze_layers:
            for param in model.parameters():
                param.requires_grad = False
        model.classifier[1] = replace_classifier(model.classifier[1].
 ↪in_features)
        model.classifier[1].apply(initialize_weights)

    else:
        raise ValueError(f"Unsupported model name: {model_name}")
    return model
```

[55]:

```
# Early stopping implementation
class EarlyStopping:
    def __init__(self, patience=5, min_delta=0):
        self.patience = patience
        self.min_delta = min_delta
        self.counter = 0
        self.best_loss = None

    def should_stop(self, val_loss):
        if self.best_loss is None or val_loss < self.best_loss - self.min_delta:
            self.best_loss = val_loss
            self.counter = 0
            return False
        else:
            self.counter += 1
            if self.counter >= self.patience:
                return True
        return False

# Train each model
def train_model_with_val(model, train_loader, val_loader, epochs, model_name):
    model = model.to(device)
    optimizer = Adam(model.parameters(), lr=1e-4)
    criterion = CrossEntropyLoss()
    early_stopping = EarlyStopping(patience=5, min_delta=0.01)
    scheduler = ReduceLROnPlateau(optimizer, mode='min', factor=0.05,␣
 ↪patience=5)

    mlflow.set_experiment("Chicken vs Duck")
    with mlflow.start_run():
        mlflow.log_param("Model", model_name)
        mlflow.log_param("Epochs", epochs)
        mlflow.log_param("Batch_Size", 16)
        mlflow.log_param("Learning_Rate", 1e-4)
        mlflow.log_param("Scheduler_Patience", 5)
        mlflow.log_param("Scheduler_Factor", 0.05)

        for epoch in range(epochs):
            start_time = time.time()
            # Training phase
            model.train()
            train_loss = 0
            for images, labels in train_loader:
                images, labels = images.to(device), labels.to(device)
                optimizer.zero_grad()
                outputs = model(images)
                loss = criterion(outputs, labels)
                loss.backward()
                optimizer.step()
                train_loss += loss.item()

            train_loss /= len(train_loader)

            # Validation phase
            model.eval()
            val_loss = 0
            correct, total = 0, 0
            with torch.no_grad():
                for images, labels in val_loader:
                    images, labels = images.to(device), labels.to(device)
                    outputs = model(images)
                    loss = criterion(outputs, labels)
                    val_loss += loss.item()
                    probabilities = softmax(outputs, dim=1)
                    _, preds = probabilities.max(1)

                    # _, preds = outputs.max(1)
                    correct += (preds == labels).sum().item()
                    total += labels.size(0)
            val_loss /= len(val_loader)
            val_accuracy = correct / total

            end_time = time.time()

            mlflow.log_metric("Train_Loss", train_loss, epoch)
            mlflow.log_metric("Val_Loss", val_loss, epoch)
            mlflow.log_metric("Val_Accuracy", val_accuracy, epoch)
            mlflow.log_metric("Epoch_Time_seconds", end_time - start_time,␣
↪epoch)  # Updated parameter name
            mlflow.log_metric("Learning_Rate", optimizer.param_groups[0]['lr'],␣
↪epoch)
            scheduler.step(val_loss)
            print(f"Epoch {epoch+1}/{epochs}, Train Loss: {train_loss/
↪len(train_loader):.4f}, "
                  f"Val Loss: {val_loss/len(val_loader):.4f}, Val Accuracy:␣
↪{correct/total:.4f}")

            # Check for early stopping
            if early_stopping.should_stop(val_loss):
                print(f"Stopping early at epoch {epoch+1}")
                break
        torch.save(model, folder_path + '/retrained_' + model_name + '.pt')
```

[56]:

```
# Train for Max. 50 epochs
N_EPOCHS = 50

for model_name in ["resnet50", "mobilenet_v2", "efficientnet_b0"]:
  model = get_model(model_name) #, freeze_layers=True
  train_model_with_val(model, train_loader, val_loader, epochs=N_EPOCHS,␣
 ↪model_name=model_name)
```

```
Epoch 1/50, Train Loss: 0.0072, Val Loss: 0.0224, Val Accuracy: 0.8696
Epoch 2/50, Train Loss: 0.0047, Val Loss: 0.0113, Val Accuracy: 0.9627
Epoch 3/50, Train Loss: 0.0034, Val Loss: 0.0133, Val Accuracy: 0.9255
Epoch 4/50, Train Loss: 0.0027, Val Loss: 0.0117, Val Accuracy: 0.9565
Epoch 5/50, Train Loss: 0.0020, Val Loss: 0.0195, Val Accuracy: 0.9130
Epoch 6/50, Train Loss: 0.0020, Val Loss: 0.0085, Val Accuracy: 0.9689
Epoch 7/50, Train Loss: 0.0014, Val Loss: 0.0158, Val Accuracy: 0.9441
Epoch 8/50, Train Loss: 0.0015, Val Loss: 0.0061, Val Accuracy: 0.9689
Epoch 9/50, Train Loss: 0.0021, Val Loss: 0.0063, Val Accuracy: 0.9689
Epoch 10/50, Train Loss: 0.0014, Val Loss: 0.0103, Val Accuracy: 0.9441
Epoch 11/50, Train Loss: 0.0011, Val Loss: 0.0118, Val Accuracy: 0.9441
Epoch 12/50, Train Loss: 0.0012, Val Loss: 0.0085, Val Accuracy: 0.9503
Epoch 13/50, Train Loss: 0.0012, Val Loss: 0.0082, Val Accuracy: 0.9689
Stopping early at epoch 13
Epoch 1/50, Train Loss: 0.0096, Val Loss: 0.0328, Val Accuracy: 0.9068
Epoch 2/50, Train Loss: 0.0064, Val Loss: 0.0219, Val Accuracy: 0.8944
Epoch 3/50, Train Loss: 0.0044, Val Loss: 0.0215, Val Accuracy: 0.9006
Epoch 4/50, Train Loss: 0.0039, Val Loss: 0.0168, Val Accuracy: 0.8944
Epoch 5/50, Train Loss: 0.0032, Val Loss: 0.0229, Val Accuracy: 0.8882
Epoch 6/50, Train Loss: 0.0027, Val Loss: 0.0163, Val Accuracy: 0.9193
Epoch 7/50, Train Loss: 0.0030, Val Loss: 0.0123, Val Accuracy: 0.9503
Epoch 8/50, Train Loss: 0.0023, Val Loss: 0.0380, Val Accuracy: 0.9255
Epoch 9/50, Train Loss: 0.0031, Val Loss: 0.0205, Val Accuracy: 0.9068
Epoch 10/50, Train Loss: 0.0024, Val Loss: 0.0071, Val Accuracy: 0.9689
Epoch 11/50, Train Loss: 0.0018, Val Loss: 0.0105, Val Accuracy: 0.9503
Epoch 12/50, Train Loss: 0.0021, Val Loss: 0.0134, Val Accuracy: 0.9441
Epoch 13/50, Train Loss: 0.0020, Val Loss: 0.0129, Val Accuracy: 0.9503
Epoch 14/50, Train Loss: 0.0018, Val Loss: 0.0099, Val Accuracy: 0.9503
Epoch 15/50, Train Loss: 0.0017, Val Loss: 0.0254, Val Accuracy: 0.8944
Stopping early at epoch 15
Epoch 1/50, Train Loss: 0.0100, Val Loss: 0.0414, Val Accuracy: 0.8882
Epoch 2/50, Train Loss: 0.0070, Val Loss: 0.0334, Val Accuracy: 0.8385
Epoch 3/50, Train Loss: 0.0052, Val Loss: 0.0206, Val Accuracy: 0.9379
Epoch 4/50, Train Loss: 0.0043, Val Loss: 0.0193, Val Accuracy: 0.9627
Epoch 5/50, Train Loss: 0.0032, Val Loss: 0.0322, Val Accuracy: 0.9379
Epoch 6/50, Train Loss: 0.0031, Val Loss: 0.0162, Val Accuracy: 0.9193
Epoch 7/50, Train Loss: 0.0031, Val Loss: 0.0377, Val Accuracy: 0.9503
Epoch 8/50, Train Loss: 0.0024, Val Loss: 0.0144, Val Accuracy: 0.9441
Epoch 9/50, Train Loss: 0.0020, Val Loss: 0.0145, Val Accuracy: 0.9503
Epoch 10/50, Train Loss: 0.0017, Val Loss: 0.0163, Val Accuracy: 0.9441
Epoch 11/50, Train Loss: 0.0024, Val Loss: 0.0220, Val Accuracy: 0.9441
Epoch 12/50, Train Loss: 0.0033, Val Loss: 0.0127, Val Accuracy: 0.9441
Epoch 13/50, Train Loss: 0.0017, Val Loss: 0.0091, Val Accuracy: 0.9627
Epoch 14/50, Train Loss: 0.0020, Val Loss: 0.0235, Val Accuracy: 0.8944
Epoch 15/50, Train Loss: 0.0015, Val Loss: 0.0165, Val Accuracy: 0.9379
Epoch 16/50, Train Loss: 0.0017, Val Loss: 0.0245, Val Accuracy: 0.9130
Epoch 17/50, Train Loss: 0.0020, Val Loss: 0.0157, Val Accuracy: 0.9379
Epoch 18/50, Train Loss: 0.0016, Val Loss: 0.0101, Val Accuracy: 0.9565
Stopping early at epoch 18
```

1.3 Model Evaluation

[57]:

```
def evaluate_model(model, test_loader):
    model.eval()
    all_preds = []
    all_labels = []

    with torch.no_grad():
        for images, labels in test_loader:
            images, labels = images.to(device), labels.to(device)
            outputs = model(images)
            probabilities = softmax(outputs, dim=1)
            _, preds = probabilities.max(1)

            all_preds.append(preds.cpu())
            all_labels.append(labels.cpu())

    all_preds = torch.cat(all_preds)
    all_labels = torch.cat(all_labels)

    print("Classification Report:")
    print(classification_report(all_labels, all_preds, target_names=["Chicken",␣
 ↪"Duck"]))

    print("Confusion Matrix:")
    print(confusion_matrix(all_labels, all_preds))


for model_name in ["resnet50", "mobilenet_v2", "efficientnet_b0"]:
  model = torch.load(folder_path + '/retrained_' + model_name + '.pt',␣
 ↪weights_only=False)
  model.to(device)
  evaluate_model(model, test_loader)
```

```
Classification Report:
              precision    recall  f1-score   support

     Chicken       0.99      0.92      0.95       172
```

```
        Duck       0.96      1.00      0.98       310

    accuracy                           0.97       482
   macro avg       0.98      0.96      0.97       482
weighted avg       0.97      0.97      0.97       482

Confusion Matrix:
[[158  14]
 [  1 309]]
Classification Report:
              precision    recall  f1-score   support

     Chicken       0.96      0.91      0.93       172
        Duck       0.95      0.98      0.96       310

    accuracy                           0.95       482
   macro avg       0.95      0.94      0.95       482
weighted avg       0.95      0.95      0.95       482

Confusion Matrix:
[[156  16]
 [  7 303]]
Classification Report:
              precision    recall  f1-score   support

     Chicken       0.93      0.91      0.92       172
        Duck       0.95      0.96      0.96       310

    accuracy                           0.95       482
   macro avg       0.94      0.94      0.94       482
weighted avg       0.95      0.95      0.95       482

Confusion Matrix:
[[157  15]
 [ 11 299]]
```

Looks like ResNet50 model gives us the best metrics

2 Part 2: Sentiment Analysis Classifier

```
[2]: folder_path = a5_data_path + '/sentiment-data'
     folder_path
```

```
[2]: './data/sentiment-data'
```

```
[5]: !ls 'drive/MyDrive/AML/Assignment 5/data/sentiment-data'
```

[6]:
```
# Load the dataset
file_path = folder_path + "/training.1600000.processed.noemoticon.csv"
data = pd.read_csv(file_path, encoding='latin-1')

df = pd.DataFrame();
df['text'] = data.iloc[:,5]
df['sentiment'] = data.iloc[:,0].apply(lambda x: 'positive' if x == 4 else␣
 ↪'neutral' if x == 2 else 'negative')
```

[7]:
```
df
```

[7]:
```
                                                       text sentiment
0        is upset that he can't update his Facebook by …  negative
1        @Kenichan I dived many times for the ball. Man…  negative
2          my whole body feels itchy and like its on fire   negative
3        @nationwideclass no, it's not behaving at all…  negative
4                            @Kwesidei not the whole crew   negative
…                                                      …        …
1048567            My GrandMa is making Dinenr with my Mum   positive
1048568  Mid-morning snack time… A bowl of cheese noo…  positive
1048569  @ShaDeLa same here  say it like from the Termi…  positive
1048570              @DestinyHope92 im great thaanks  wbuu?  positive
1048571                cant wait til her date this weekend   positive

[1048572 rows x 2 columns]
```

[8]:
```
# Encode sentiment labels
encoder = LabelEncoder()
df['label'] = encoder.fit_transform(df['sentiment'])

# split train dataset into train, validation and test sets
train_text, temp_text, train_labels, temp_labels = train_test_split(df['text'],␣
 ↪df['label'],
                                                                                ␣
 ↪random_state=2018,
                                                                                 test_size=0.
 ↪3,
                                                                                ␣
 ↪stratify=df['label'])


val_text, test_text, val_labels, test_labels = train_test_split(temp_text,␣
 ↪temp_labels,
                                                                            ␣
 ↪random_state=2018,
                                                                             test_size=0.5,
                                                                                ␣
↪stratify=temp_labels)
```

[9]:

```
# import BERT-base pretrained model
bert = AutoModel.from_pretrained('bert-base-uncased')

# Load the BERT tokenizer
tokenizer = BertTokenizerFast.from_pretrained('bert-base-uncased')
```

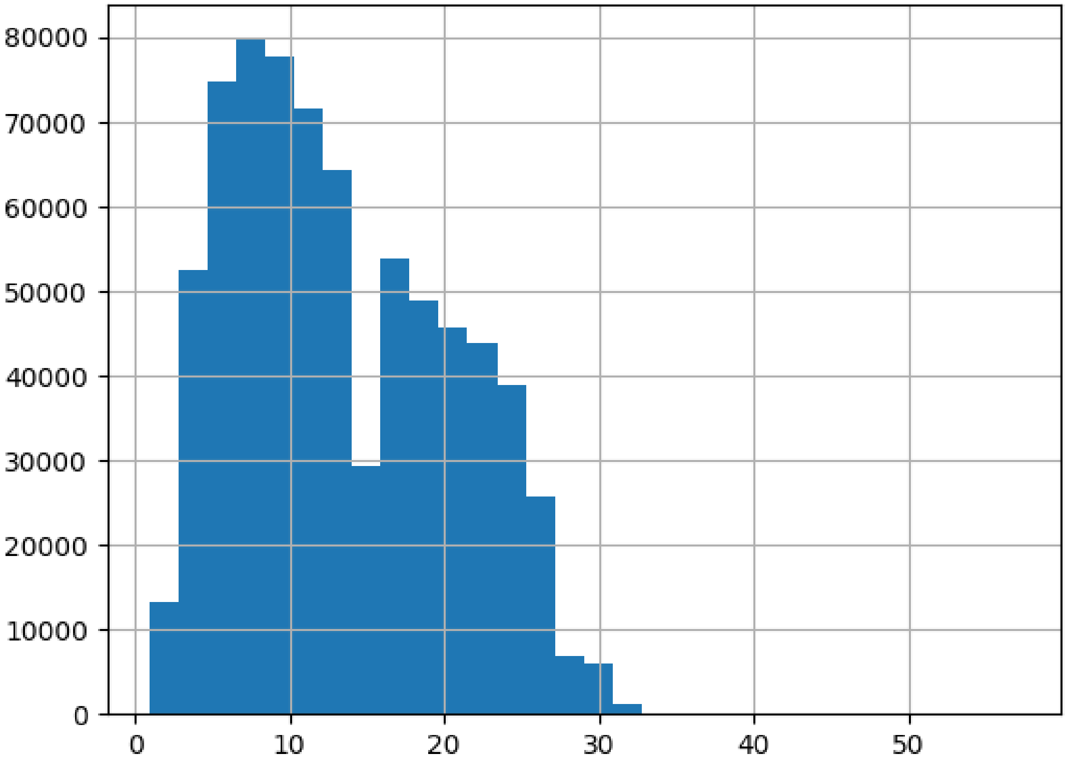
2.0.1 Tokenize Sentences

[10]:

```
# get length of all the messages in the train set
seq_len = [len(i.split()) for i in train_text]

pd.Series(seq_len).hist(bins = 30)
```

[10]:

```
<Axes: >
```



Setting padding length to 30

[11]:

```
# tokenize and encode sequences
tokens_train = tokenizer.batch_encode_plus(
    train_text.tolist(),
    max_length = 30,
    padding=True,
    truncation=True
)
tokens_val = tokenizer.batch_encode_plus(
    val_text.tolist(),
    max_length = 30,
    padding=True,
    truncation=True
)
tokens_test = tokenizer.batch_encode_plus(
    test_text.tolist(),
    max_length = 30,
    padding=True,
    truncation=True
)
```

[12]:

```
## convert lists to tensors
train_seq = torch.tensor(tokens_train['input_ids'])
train_mask = torch.tensor(tokens_train['attention_mask'])
train_y = torch.tensor(train_labels.tolist())

val_seq = torch.tensor(tokens_val['input_ids'])
val_mask = torch.tensor(tokens_val['attention_mask'])
val_y = torch.tensor(val_labels.tolist())

test_seq = torch.tensor(tokens_test['input_ids'])
test_mask = torch.tensor(tokens_test['attention_mask'])
test_y = torch.tensor(test_labels.tolist())
```

[13]:

```
batch_size = 512
train_data = TensorDataset(train_seq, train_mask, train_y)
train_sampler = RandomSampler(train_data)
train_dataloader = DataLoader(train_data, sampler=train_sampler,␣
 ↪batch_size=batch_size)

val_data = TensorDataset(val_seq, val_mask, val_y)
val_sampler = SequentialSampler(val_data)
val_dataloader = DataLoader(val_data, sampler = val_sampler,␣
 ↪batch_size=batch_size)
```

[14]:

```
# freeze all the parameters
for param in bert.parameters():
    param.requires_grad = False
```

[15]:

```
class BERT_Arch(nn.Module):
    def __init__(self, bert):
      super(BERT_Arch, self).__init__()
      self.bert = bert
      self.dropout = nn.Dropout(0.1)
      self.relu =  nn.ReLU()
      self.fc1 = nn.Linear(768,128)
      self.fc2 = nn.Linear(128,2)
      self.softmax = nn.LogSoftmax(dim=1)

    #define the forward pass
    def forward(self, sent_id, mask):
      _, cls_hs = self.bert(sent_id, attention_mask=mask,return_dict=False)
      x = self.fc1(cls_hs)
      x = self.relu(x)
      x = self.dropout(x)
      x = self.fc2(x)
      x = self.softmax(x)
      return x
```

[16]:

```
# pass the pre-trained BERT to our define architecture
model = BERT_Arch(bert)

# push the model to GPU
model = model.to(device)
optimizer = Adam(model.parameters(), lr=1e-4)
```

[17]:

```
#compute the class weights
class_weights = compute_class_weight("balanced", classes=train_labels.unique(),␣
 ↪y=train_labels)
print("Class Weights:",class_weights)
```

```
Class Weights: [0.65536065 2.10915904]
```

[18]:

```
# converting list of class weights to a tensor
weights= torch.tensor(class_weights,dtype=torch.float)

# push to GPU
weights = weights.to(device)

# define the loss function
cross_entropy  = nn.NLLLoss(weight=weights)

# number of training epochs
epochs = 10
```

[23]:
```
# function to train the model
def train():
  model.train()
  total_loss, total_accuracy = 0, 0
  total_preds=[]
  for step,batch in enumerate(train_dataloader):
    # progress update after every 50 batches.
    if step % 50 == 0 and not step == 0:
      print('  Batch {:>5,}  of  {:>5,}.'.format(step, len(train_dataloader)))

    batch = [r.to(device) for r in batch]
    sent_id, mask, labels = batch

    model.zero_grad()
    preds = model(sent_id, mask)
    loss = cross_entropy(preds, labels)
    total_loss = total_loss + loss.item()
    loss.backward()
    torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
    optimizer.step()
    preds=preds.detach().cpu().numpy()
    total_preds.append(preds)

  avg_loss = total_loss / len(train_dataloader)

  # predictions are in the form of (no. of batches, size of batch, no. of␣
 ↪classes).
  # reshape the predictions in form of (number of samples, no. of classes)
  total_preds  = np.concatenate(total_preds, axis=0)

  return avg_loss, total_preds


# function for evaluating the model
def evaluate():
  print("\nEvaluating...")
  model.eval()
  total_loss, total_accuracy = 0, 0
  total_preds = []

  for step,batch in enumerate(val_dataloader):
    if step % 50 == 0 and not step == 0:
      print('  Batch {:>5,}  of  {:>5,}.'.format(step, len(val_dataloader)))

    batch = [t.to(device) for t in batch]
    sent_id, mask, labels = batch
    with torch.no_grad():
      preds = model(sent_id, mask)
      loss = cross_entropy(preds,labels)
      total_loss = total_loss + loss.item()
      preds = preds.detach().cpu().numpy()
      total_preds.append(preds)

  avg_loss = total_loss / len(val_dataloader)
  total_preds = np.concatenate(total_preds, axis=0)

  return avg_loss, total_preds
```

[24]:
```
best_valid_loss = float('inf')

train_losses=[]
valid_losses=[]

for epoch in range(epochs):
    print('\n Epoch {:} / {:}'.format(epoch + 1, epochs))

    train_loss, _ = train()
    valid_loss, _ = evaluate()

    if valid_loss < best_valid_loss:
        best_valid_loss = valid_loss
        torch.save(model.state_dict(), folder_path + '/saved_weights.pt')

    train_losses.append(train_loss)
    valid_losses.append(valid_loss)

    print(f'\nTraining Loss: {train_loss:.3f}')
    print(f'Validation Loss: {valid_loss:.3f}')
```

```

Epoch 1 / 10
 Batch    50  of  1,434.
 Batch   100  of  1,434.
 Batch   150  of  1,434.
 Batch   200  of  1,434.
 Batch   250  of  1,434.
 Batch   300  of  1,434.
 Batch   350  of  1,434.
 Batch   400  of  1,434.
 Batch   450  of  1,434.
 Batch   500  of  1,434.
 Batch   550  of  1,434.
 Batch   600  of  1,434.
 Batch   650  of  1,434.
 Batch   700  of  1,434.
```

```
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.583
Validation Loss: 0.526

 Epoch 2 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.534
Validation Loss: 0.502

 Epoch 3 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.
```

```
Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.523
Validation Loss: 0.498

 Epoch 4 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.518
Validation Loss: 0.489

 Epoch 5 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.515
Validation Loss: 0.487

 Epoch 6 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.512
Validation Loss: 0.486

 Epoch 7 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.510
Validation Loss: 0.492

 Epoch 8 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.509
Validation Loss: 0.481

 Epoch 9 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.

Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.507
Validation Loss: 0.487

 Epoch 10 / 10
  Batch    50  of  1,434.
  Batch   100  of  1,434.
  Batch   150  of  1,434.
  Batch   200  of  1,434.
  Batch   250  of  1,434.
  Batch   300  of  1,434.
  Batch   350  of  1,434.
  Batch   400  of  1,434.
  Batch   450  of  1,434.
  Batch   500  of  1,434.
  Batch   550  of  1,434.
  Batch   600  of  1,434.
  Batch   650  of  1,434.
  Batch   700  of  1,434.
  Batch   750  of  1,434.
  Batch   800  of  1,434.
  Batch   850  of  1,434.
  Batch   900  of  1,434.
  Batch   950  of  1,434.
  Batch 1,000  of  1,434.
  Batch 1,050  of  1,434.
  Batch 1,100  of  1,434.
  Batch 1,150  of  1,434.
  Batch 1,200  of  1,434.
  Batch 1,250  of  1,434.
  Batch 1,300  of  1,434.
  Batch 1,350  of  1,434.
  Batch 1,400  of  1,434.
```

```
Evaluating…
  Batch    50  of    308.
  Batch   100  of    308.
  Batch   150  of    308.
  Batch   200  of    308.
  Batch   250  of    308.
  Batch   300  of    308.

Training Loss: 0.506
Validation Loss: 0.490
```

[ ]:

```
#load weights of best model
path = folder_path + '/saved_weights.pt'
model.load_state_dict(torch.load(path))
```

[27]:

```
test_data = TensorDataset(test_seq, test_mask, test_y)
test_sampler = SequentialSampler(test_data)
test_dataloader = DataLoader(test_data, sampler = test_sampler,␣
  ↪batch_size=batch_size)
```

[29]:

```
# get predictions for test data

model.eval()
total_loss, total_accuracy = 0, 0
total_preds = []

for step,batch in enumerate(test_dataloader):
    batch = [t.to(device) for t in batch]
    sent_id, mask, labels = batch
    with torch.no_grad():
      preds = model(sent_id, mask)
      loss = cross_entropy(preds,labels)
      total_loss = total_loss + loss.item()
      preds = preds.detach().cpu().numpy()
      preds = np.argmax(preds, axis = 1)
      total_preds.append(preds)

avg_loss = total_loss / len(test_dataloader)
total_preds = np.concatenate(total_preds, axis=0)
print(classification_report(test_y, total_preds))
```

```
              precision    recall  f1-score   support

           0       0.94      0.67      0.78    120000
           1       0.44      0.86      0.58     37286

    accuracy                           0.71    157286
   macro avg       0.69      0.76      0.68    157286
```

```
weighted avg       0.82      0.71      0.73    157286
```

[ ]:

This notebook was converted with convert.ploomber.io