07_model_evaluation

February 12, 2024

[1]:
```
import os
```

[2]:
```
%pwd
```

[2]:
```
'D:\\Desktop\\Deep Learning\\Lab 3\\Main MNSIT-MLPClassifer\\Research'
```

[3]:
```
os.chdir("../")
```

[4]:
```
%pwd
```

[4]:
```
'D:\\Desktop\\Deep Learning\\Lab 3\\Main MNSIT-MLPClassifer'
```

[5]:
```
import warnings

# Disable all warnings
warnings.filterwarnings("ignore")
warnings.filterwarnings("ignore", message="From .*: The name tf.
 ↪get_default_graph is deprecated.")
warnings.filterwarnings("ignore", message="From .*: The name tf.train.Optimizer␣
 ↪is deprecated.")
warnings.filterwarnings("ignore", message="From .*: The name tf.ragged.
 ↪RaggedTensorValue is deprecated.")
warnings.filterwarnings("ignore", message="From .*: The name tf.
 ↪executing_eagerly_outside_functions is deprecated.")
```

[6]:
```
import logging
import joblib
import os
from dataclasses import dataclass
from pathlib import Path
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score, precision_recall_fscore_support
from tensorflow.keras.models import load_model

# Configure logging
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s -␣
 ↪%(message)s')
```

```python
@dataclass(frozen=True)
class DataTransformationConfig:
    root_dir: Path
    X_test_file: Path
    y_test_file: Path
    scaler_file: Path
    label_encoder_file: Path
    keras_mnist_model_file: Path

class ConfigurationManager:
    def __init__(self):
        self.root_dir = Path(os.getcwd())
        self.X_test_file = self.root_dir / "dataset/Modeltraining/X_test.csv"
        self.y_test_file = self.root_dir / "dataset/Modeltraining/y_test.csv"
        self.experiment_results_dir = self.root_dir / "ModelExperiments"
        self.scaler_file = self.experiment_results_dir / "scaler.pkl"
        self.label_encoder_file = self.experiment_results_dir / "label_encoder.
 ↪pkl"
        self.keras_mnist_model_file = self.experiment_results_dir /␣
 ↪"model_nodes_64_layers_16_epochs_30.keras"

    def get_data_transformation_config(self) -> DataTransformationConfig:
        return DataTransformationConfig(
            root_dir=self.root_dir,
            X_test_file=self.X_test_file,
            y_test_file=self.y_test_file,
            scaler_file=self.scaler_file,
            label_encoder_file=self.label_encoder_file,
            keras_mnist_model_file=self.keras_mnist_model_file
        )

class ModelEvaluation:
    def __init__(self, config: DataTransformationConfig):
        self.config = config
        self.scaler = joblib.load(config.scaler_file)
        self.label_encoder = joblib.load(config.label_encoder_file)
        self.model = load_model(config.keras_mnist_model_file)

    def load_test_data(self):
        X_test = pd.read_csv(self.config.X_test_file)
        y_test = pd.read_csv(self.config.y_test_file)
        logging.info("X_test and y_test loaded")

        X_test_scaled = self.scaler.transform(X_test)
        y_test_encoded = self.label_encoder.transform(y_test)
```

```
        return X_test_scaled, y_test_encoded

    def generate_prediction_from_test(self):
        X_test_scaled, y_test_encoded = self.load_test_data()
        y_pred = self.model.predict(X_test_scaled)

        # Assuming your model outputs probabilities
        y_pred_labels = np.argmax(y_pred, axis=1)

        # Ensure y_pred_labels is of integer type for indexing
        y_pred_labels = y_pred_labels.astype(int)  # Ensure integer type

        # Initialize y_test_labels and y_pred_labels for scope
        y_test_labels = None
        y_pred_transformed = None

        try:
            # The inverse transform should be applied to y_test, not␣
↪y_pred_labels,
            # because y_test is what we compare against. We got this wrong in␣
↪the previous code.
            # It's y_test that needs to be inverse transformed if it was label␣
↪encoded.
            y_test_labels = y_test_encoded

            # Only inverse transform y_pred_labels if necessary. If your model␣
↪outputs
            # labels in the same format as y_test, you may not need to inverse␣
↪transform.
            y_pred_transformed = self.label_encoder.
↪inverse_transform(y_pred_labels)
        except Exception as e:
            logging.error(f"Error during inverse transformation: {e}")
            # Return or handle error appropriately
            return

        # Proceed with evaluation using the correctly transformed labels
        accuracy = accuracy_score(y_test_labels, y_pred_transformed)
        precision, recall, fscore, _ =␣
↪precision_recall_fscore_support(y_test_labels, y_pred_transformed,␣
↪average='macro', zero_division=0)

        logging.info(f"Accuracy: {accuracy}")
        logging.info(f"Precision: {precision}")
        logging.info(f"Recall: {recall}")
        logging.info(f"F1 Score: {fscore}")
```

```
def main():
    try:
        config_manager = ConfigurationManager()
        data_transformation_config = config_manager.
 ↪get_data_transformation_config()
        model_evaluation = ModelEvaluation(config=data_transformation_config)

        model_evaluation.generate_prediction_from_test()

    except Exception as e:
        logging.error(f"Error occurred: {e}")
        raise e

if __name__ == "__main__":
    main()
```

```
WARNING:tensorflow:From D:\Desktop\Deep Learning\Lab 2\MNSIT-
MLPClassifer\venv\lib\site-packages\keras\src\losses.py:2976: The name
tf.losses.sparse_softmax_cross_entropy is deprecated. Please use
tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

WARNING:tensorflow:From D:\Desktop\Deep Learning\Lab 2\MNSIT-
MLPClassifer\venv\lib\site-packages\keras\src\backend.py:1398: The name
tf.executing_eagerly_outside_functions is deprecated. Please use
tf.compat.v1.executing_eagerly_outside_functions instead.


2024-02-12 21:59:44,345 - WARNING - From D:\Desktop\Deep Learning\Lab 2\MNSIT-
MLPClassifer\venv\lib\site-packages\keras\src\backend.py:1398: The name
tf.executing_eagerly_outside_functions is deprecated. Please use
tf.compat.v1.executing_eagerly_outside_functions instead.


WARNING:tensorflow:From D:\Desktop\Deep Learning\Lab 2\MNSIT-
MLPClassifer\venv\lib\site-packages\keras\src\optimizers\__init__.py:309: The
name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer
instead.


2024-02-12 21:59:45,066 - WARNING - From D:\Desktop\Deep Learning\Lab 2\MNSIT-
MLPClassifer\venv\lib\site-packages\keras\src\optimizers\__init__.py:309: The
name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer
instead.

2024-02-12 21:59:47,048 - INFO - X_test and y_test loaded

395/395 [==============================] - 2s 4ms/step

2024-02-12 21:59:49,682 - INFO - Accuracy: 0.9651512751465231
```

```
2024-02-12 21:59:49,686 - INFO - Precision: 0.9653371088522127
2024-02-12 21:59:49,689 - INFO - Recall: 0.9652626971367433
2024-02-12 21:59:49,695 - INFO - F1 Score: 0.965256306308318
```

[ ]: