Module 4: Critical Thinking Option 1

Jocelyn Strmec

Colorado State University Global

CSC580 Applying Machine Learning and Neural Networks - Capstone

Pubali Banerjee

09/07/2025

Toxicology testing is a binary classification problem, something is either toxic or non-toxic. This means that when evaluating performance of a toxicology model metrics that properly capture the nuance of an imbalanced dataset as Tox21 is skewed and has many more non-toxic than toxic samples. F1 score will balance precision of how many predicted toxic molecules are truly toxic and recall of how many of the actual toxic molecules are correctly identified so that false negatives or false positives are penalized. ROC-AUC (Area Under Receiver Operating Characteristic Curve) will be an additionally beneficial metric of measurement because it can help distinguish between toxic and non-toxic across thresholds and will be less sensitive to class imbalance. Additionally ROC-AUC is widely used in cheminformatics and should be a reasonable indicator of model performance (Cesar et al., 2017).

Metric	Score
Accuracy	0.9616
F1-score	0.1176
ROC-AUC	0.7861

Simply looking at accuracy the model performance would seem excellent, however given that most of the data is non-toxic it would be quite easy to always guess non-toxic and get a high accuracy. The F1-score is very low because the model isn't balancing precision and recall, meaning it is failing to detect toxic compounds and is predicting non-toxic too much. The final metric, ROC-AUC, proves the model can separate toxic from non-toxic better than chance (0.5), but it isn't perfect. The gap between high ROC-AUC and low F1 means the threshold 0.5 is not optimal. If I shift the threshold to 0.2, F1 will likely improve.

Considering the training and validation losses there is a steady drop across epochs, reaching ~0.08 by epoch 9, but validation loss decreases during the first few epochs, then flattens around epoch 4–5 at ~0.14. This suggests that the model is overfitting. The model keeps memorizing training examples but struggles to generalize. Because overfitting starts after ~4–5 epochs, adding early stopping would help in addition to a bit of regularization by increasing dropout rate and adding L2 weight regularization and using oversampling to get more positives (toxic) with SMOTE or weighted sampling. All these combined should increase the F1 score and ROC-AUC.

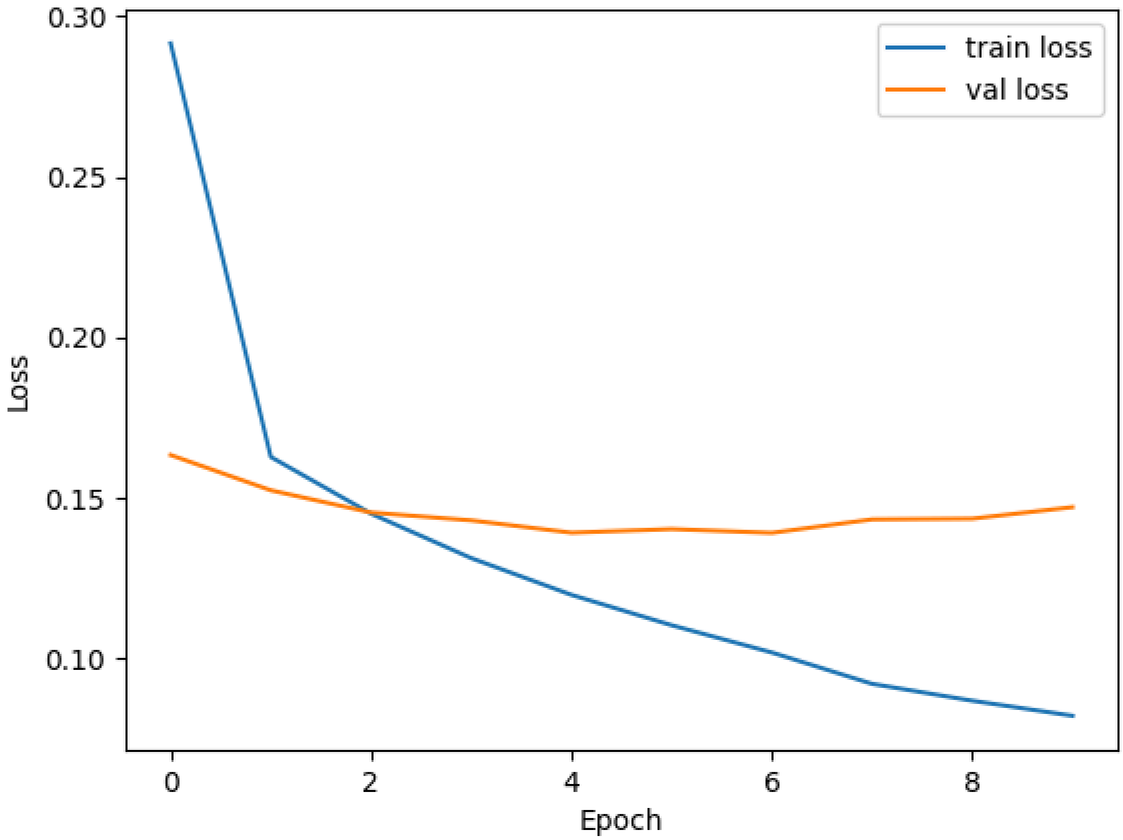
When applied the changes make a positive impact on the F1-score, although lower the other metrics showing an increase in labeling items as toxic.

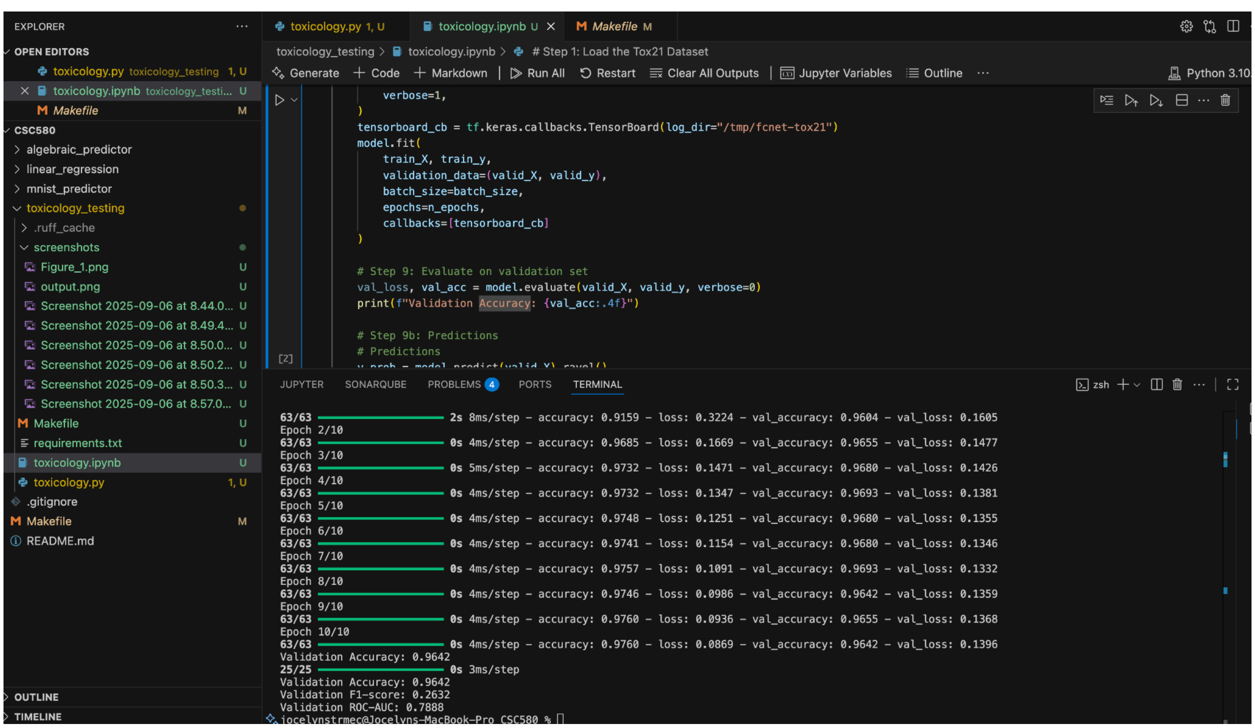
Metric	Score
Accuracy	0.9399
F1-score	0.3562
ROC-AUC	0.7661

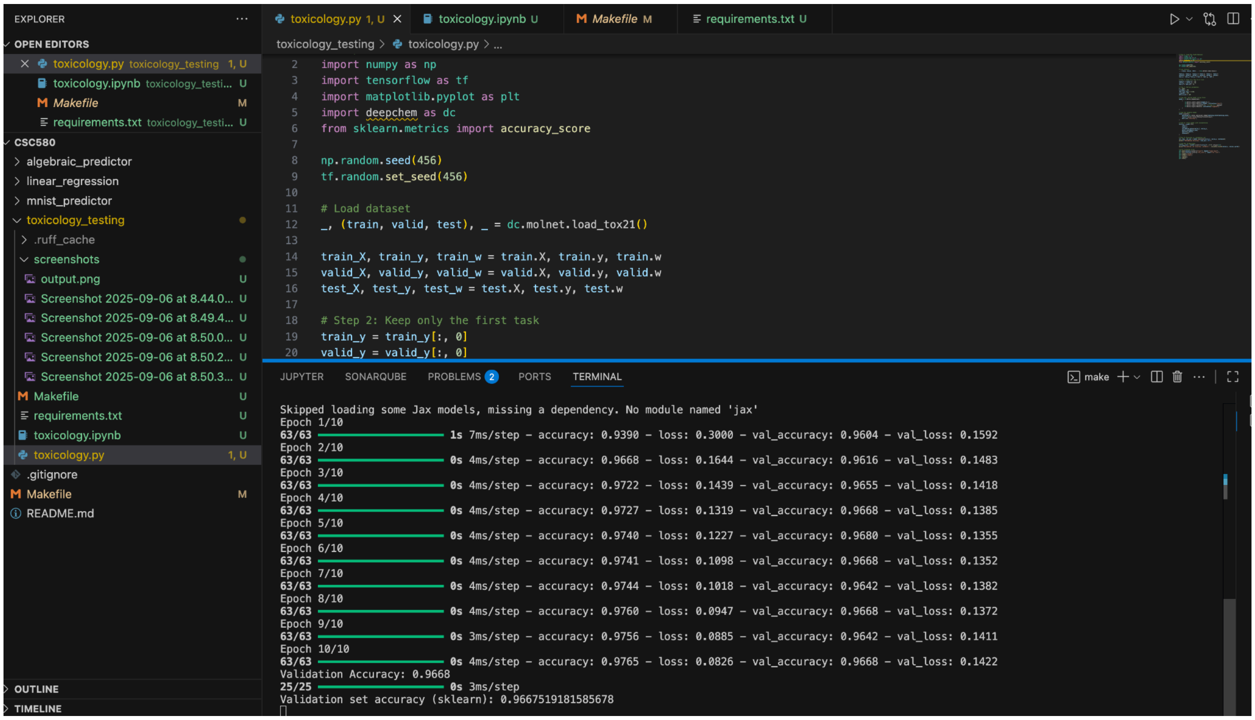
Through a few iterations of adjustment we could likely continue to balance the scores but for toxicology testing exercise sufficient improvement has been made to showcase how knowing data, interpreting loss functions and metrics can help make a more robust model.

References

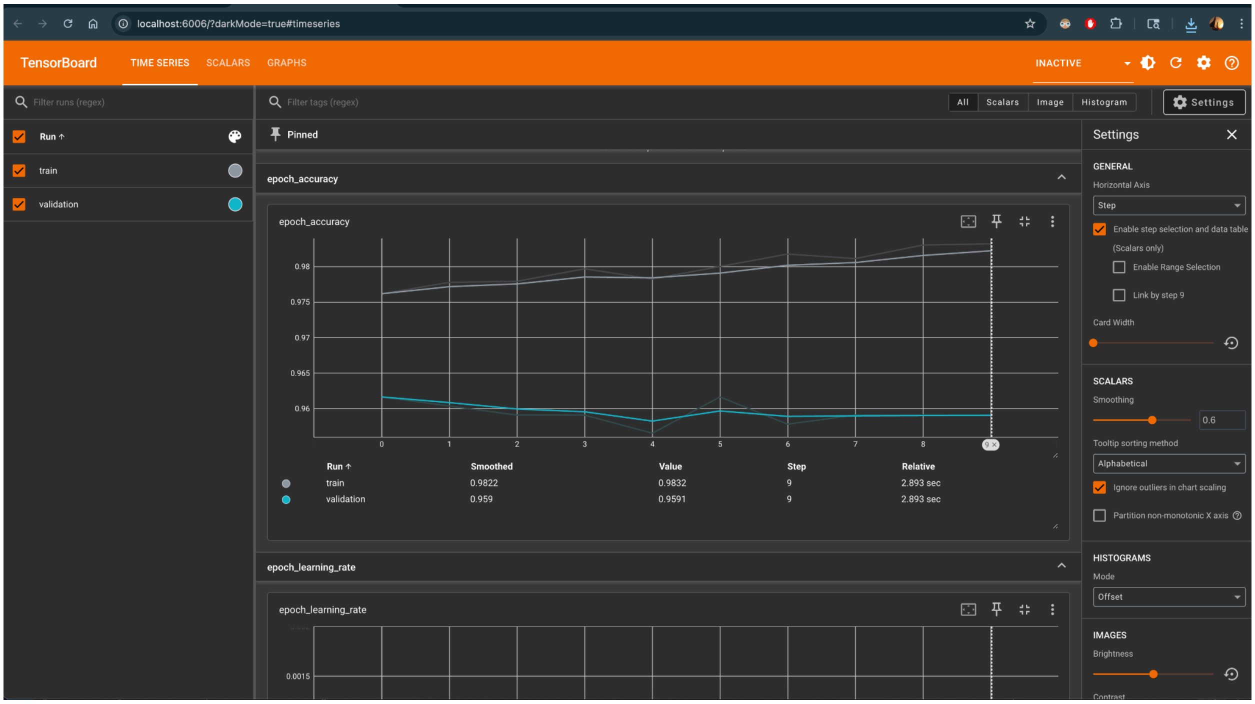
Cesar, J., Santos, Andrelly Martins-José, Koen Augustyns, & Winter, H. D. (2017). The power metric: a new statistically robust enrichment-type metric for virtual screening applications with early recovery capability. *Journal of Cheminformatics*, *9*(1). https://doi.org/10.1186/s13321-016-0189-4

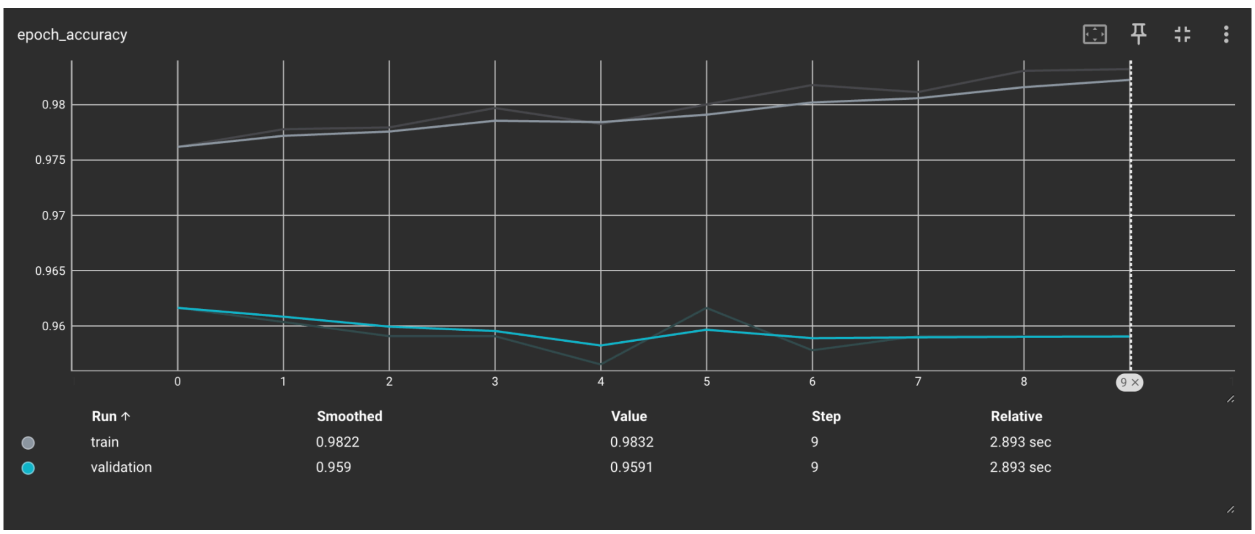

train loss
val loss
0.30
0.25
0.20
0.15
0.10
Loss
0
2
4
6
8
Epoch

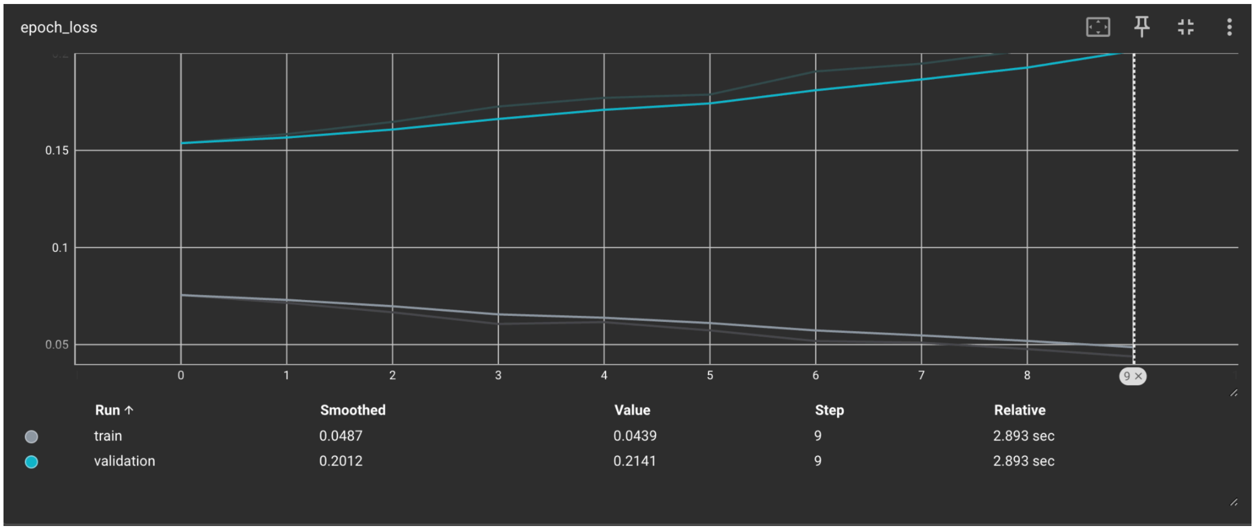

EXPLORER
OPEN EDITORS
toxicology.py toxicology_testing 1, U
toxicology.ipynb toxicology_testi... U
Makefile M
CSC580
algebraic_predictor
linear_regression
mnist_predictor
toxicology_testing
.ruff_cache
screenshots
Figure_1.png U
output.png U
Screenshot 2025-09-06 at 8.44.0... U
Screenshot 2025-09-06 at 8.49.4... U
Screenshot 2025-09-06 at 8.50.0... U
Screenshot 2025-09-06 at 8.50.2... U
Screenshot 2025-09-06 at 8.50.3... U
Screenshot 2025-09-06 at 8.57.0... U
Makefile U
requirements.txt U
toxicology.ipynb U
toxicology.py 1, U
.gitignore
Makefile M
README.md
OUTLINE
TIMELINE
toxicology.py 1, U
toxicology.ipynb U
Makefile M
toxicology_testing > toxicology.ipynb > # Step 1: Load the Tox21 Dataset
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
Python 3.10
verbose=1,
)
tensorboard_cb = tf.keras.callbacks.TensorBoard(log_dir="/tmp/fcnet-tox21")
model.fit(
train_X, train_y,
validation_data=(valid_X, valid_y),
batch_size=batch_size,
epochs=n_epochs,
callbacks=[tensorboard_cb]
)
Step 9: Evaluate on validation set
val_loss, val_acc = model.evaluate(valid_X, valid_y, verbose=0)
print(f"Validation Accuracy: {val_acc:.4f}")
# Step 9b: Predictions
# Predictions
[2]
JUPYTER
SONARQUBE
PROBLEMS 4
PORTS
TERMINAL
zsh
63/63 2s 8ms/step - accuracy: 0.9159 - loss: 0.3224 - val_accuracy: 0.9604 - val_loss: 0.1605
Epoch 2/10
63/63 0s 4ms/step - accuracy: 0.9685 - loss: 0.1669 - val_accuracy: 0.9655 - val_loss: 0.1477
Epoch 3/10
63/63 0s 5ms/step - accuracy: 0.9732 - loss: 0.1471 - val_accuracy: 0.9680 - val_loss: 0.1426
Epoch 4/10
63/63 0s 4ms/step - accuracy: 0.9732 - loss: 0.1347 - val_accuracy: 0.9693 - val_loss: 0.1381
Epoch 5/10
63/63 0s 4ms/step - accuracy: 0.9748 - loss: 0.1251 - val_accuracy: 0.9680 - val_loss: 0.1355
Epoch 6/10
63/63 0s 4ms/step - accuracy: 0.9741 - loss: 0.1154 - val_accuracy: 0.9680 - val_loss: 0.1346
Epoch 7/10
63/63 0s 4ms/step - accuracy: 0.9757 - loss: 0.1091 - val_accuracy: 0.9693 - val_loss: 0.1332
Epoch 8/10
63/63 0s 4ms/step - accuracy: 0.9746 - loss: 0.0986 - val_accuracy: 0.9642 - val_loss: 0.1359
Epoch 9/10
63/63 0s 4ms/step - accuracy: 0.9760 - loss: 0.0936 - val_accuracy: 0.9655 - val_loss: 0.1368
Epoch 10/10
63/63 0s 4ms/step - accuracy: 0.9760 - loss: 0.0869 - val_accuracy: 0.9642 - val_loss: 0.1396
Validation Accuracy: 0.9642
25/25 0s 3ms/step
Validation Accuracy: 0.9642
Validation F1-score: 0.2632
Validation ROC-AUC: 0.7888
jocelynstrmec@Jocelyns-MacBook-Pro CSC580 %

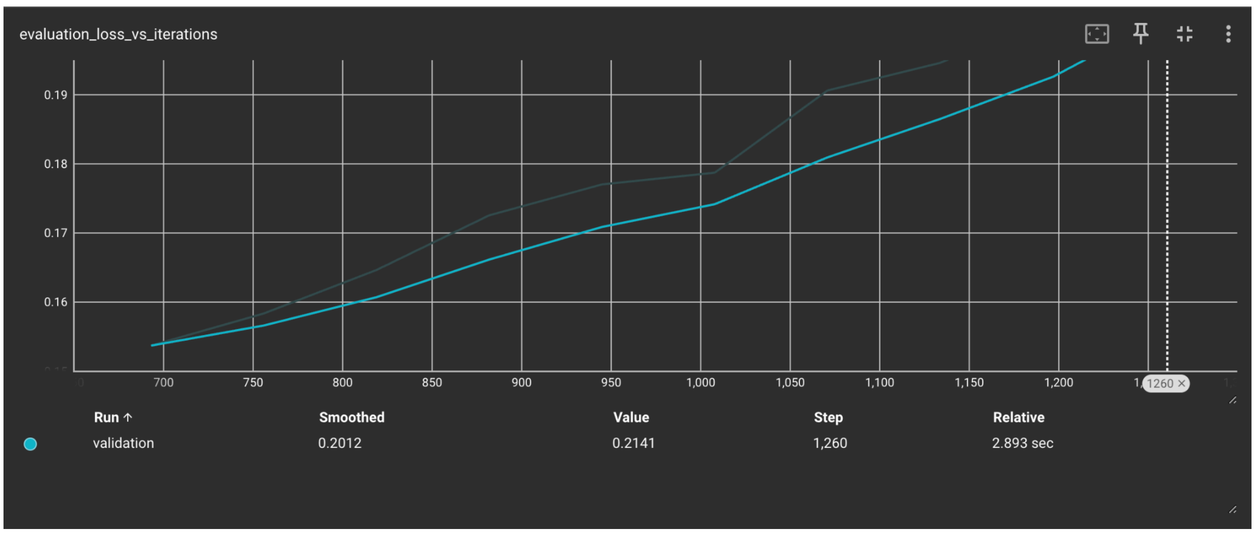


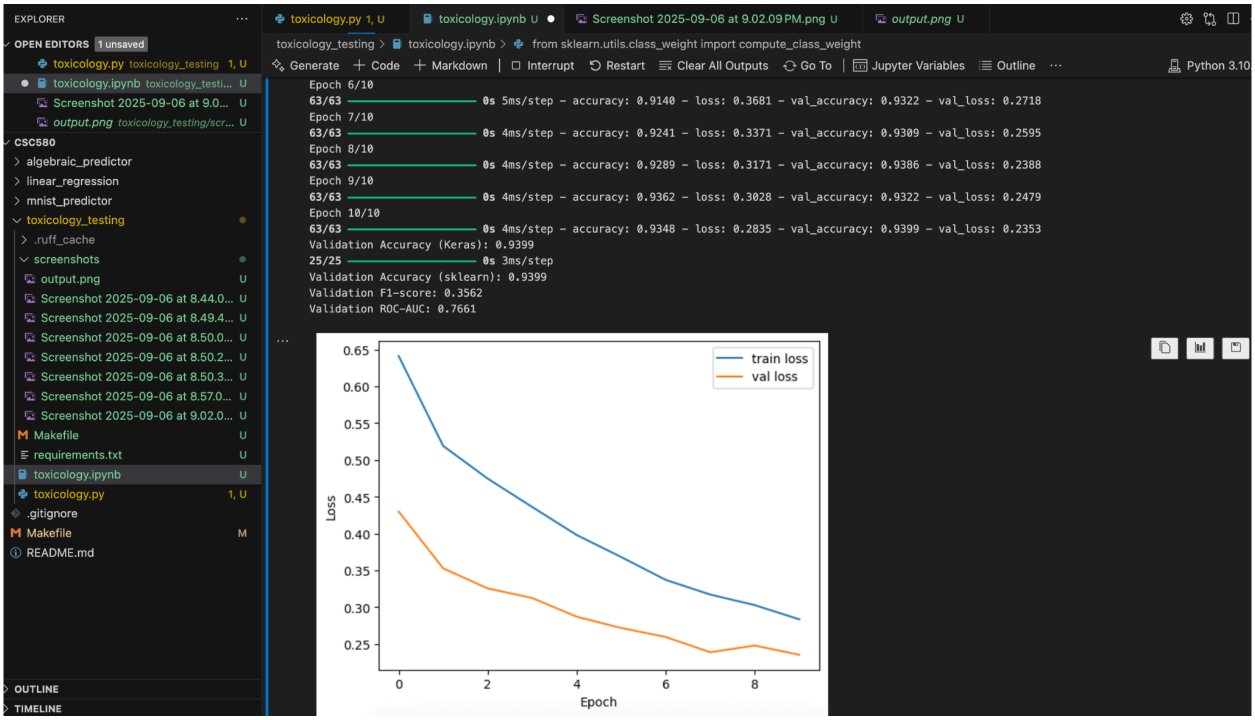
EXPLORER
OPEN EDITORS
toxicology.py toxicology_testing 1, U
toxicology.ipynb toxicology_testi... U
Makefile M
requirements.txt toxicology_testi... U
CSC580
algebraic_predictor
linear_regression
mnist_predictor
toxicology_testing
.ruff_cache
screenshots
output.png U
Makefile U
requirements.txt U
toxicology.ipynb U
toxicology.py 1, U
.gitignore
Makefile M
README.md
OUTLINE
TIMELINE
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import deepchem as dc
from sklearn.metrics import accuracy_score
np.random.seed(456)
tf.random.set_seed(456)
# Load dataset
_, (train, valid, test), _ = dc.molnet.load_tox21()
train_X, train_y, train_w = train.X, train.y, train.w
valid_X, valid_y, valid_w = valid.X, valid.y, valid.w
test_X, test_y, test_w = test.X, test.y, test.w
# Step 2: Keep only the first task
train_y = train_y[:, 0]
valid_y = valid_y[:, 0]
JUPYTER
SONARQUBE
PROBLEMS 2
PORTS
TERMINAL
Skipped loading some Jax models, missing a dependency. No module named 'jax'
Epoch 1/10
63/63 1s 7ms/step - accuracy: 0.9390 - loss: 0.3000 - val_accuracy: 0.9604 - val_loss: 0.1592
Epoch 2/10
63/63 0s 4ms/step - accuracy: 0.9668 - loss: 0.1644 - val_accuracy: 0.9616 - val_loss: 0.1483
Epoch 3/10
63/63 0s 4ms/step - accuracy: 0.9722 - loss: 0.1439 - val_accuracy: 0.9655 - val_loss: 0.1418
Epoch 4/10
63/63 0s 4ms/step - accuracy: 0.9727 - loss: 0.1319 - val_accuracy: 0.9668 - val_loss: 0.1385
Epoch 5/10
63/63 0s 4ms/step - accuracy: 0.9740 - loss: 0.1227 - val_accuracy: 0.9680 - val_loss: 0.1355
Epoch 6/10
63/63 0s 4ms/step - accuracy: 0.9741 - loss: 0.1098 - val_accuracy: 0.9668 - val_loss: 0.1352
Epoch 7/10
63/63 0s 4ms/step - accuracy: 0.9744 - loss: 0.1018 - val_accuracy: 0.9642 - val_loss: 0.1382
Epoch 8/10
63/63 0s 4ms/step - accuracy: 0.9760 - loss: 0.0947 - val_accuracy: 0.9668 - val_loss: 0.1372
Epoch 9/10
63/63 0s 3ms/step - accuracy: 0.9756 - loss: 0.0885 - val_accuracy: 0.9642 - val_loss: 0.1411
Epoch 10/10
63/63 0s 4ms/step - accuracy: 0.9765 - loss: 0.0826 - val_accuracy: 0.9668 - val_loss: 0.1422
Validation Accuracy: 0.9668
25/25 0s 3ms/step
Validation set accuracy (sklearn): 0.9667519181585678

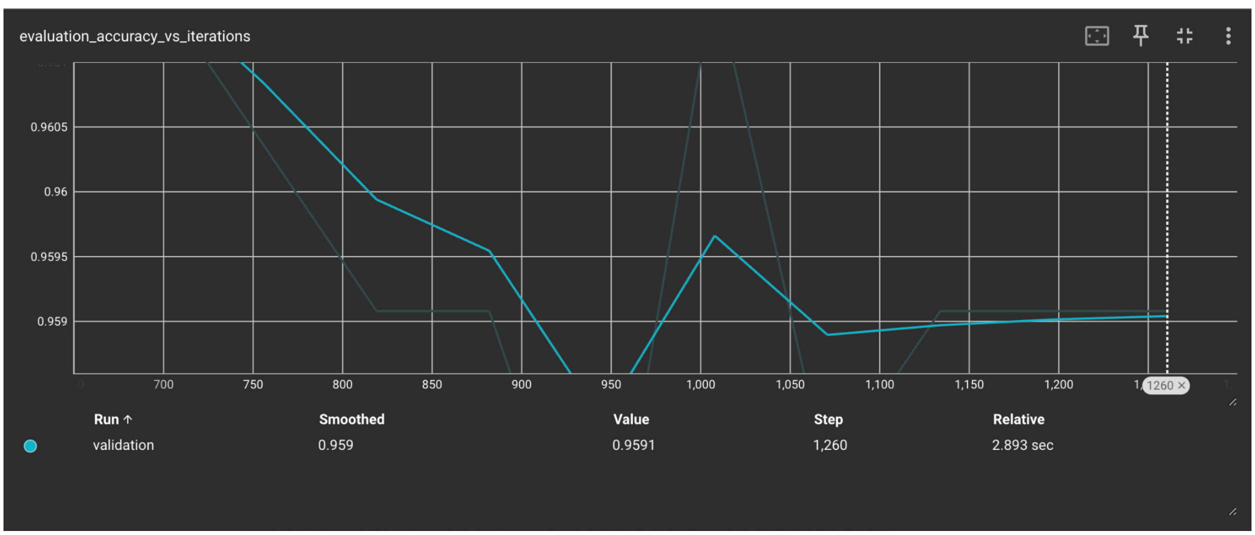

localhost:6006/?darkMode=true#timeseries
TensorBoard
TIME SERIES
SCALARS
GRAPHS
INACTIVE
Filter runs (regex)
Run ↑
train
validation
Filter tags (regex)
All
Scalars
Image
Histogram
Settings
Pinned
epoch_accuracy
0.98
0.975
0.97
0.965
0.96
0 1 2 3 4 5 6 7 8 9
Run ↑ Smoothed Value Step Relative
train 0.9822 0.9832 9 2.893 sec
validation 0.959 0.9591 9 2.893 sec
epoch_learning_rate
0.0015
Settings
GENERAL
Horizontal Axis
Step
Enable step selection and data table
(Scalars only)
Enable Range Selection
Link by step 9
Card Width
SCALARS
Smoothing
0.6
Tooltip sorting method
Alphabetical
Ignore outliers in chart scaling
Partition non-monotonic X axis
HISTOGRAMS
Mode
Offset
IMAGES
Brightness
Contrast


epoch_accuracy
0.98
0.975
0.97
0.965
0.96
0 1 2 3 4 5 6 7 8 9
Run ↑ Smoothed Value Step Relative
train 0.9822 0.9832 9 2.893 sec
validation 0.959 0.9591 9 2.893 sec


epoch_loss
0.15
0.1
0.05
0
1
2
3
4
5
6
7
8
9
Run
Smoothed
Value
Step
Relative
train
0.0487
0.0439
9
2.893 sec
validation
0.2012
0.2141
9
2.893 sec


evaluation_loss_vs_iterations
0.19
0.18
0.17
0.16
700
750
800
850
900
950
1,000
1,050
1,100
1,150
1,200
1260
Run
Smoothed
Value
Step
Relative
validation
0.2012
0.2141
1,260
2.893 sec


EXPLORER
OPEN EDITORS 1 unsaved
toxicology.py toxicology_testing 1, U
toxicology.ipynb toxicology_testi... U
Screenshot 2025-09-06 at 9.0... U
output.png toxicology_testing/scr... U
CSC580
algebraic_predictor
linear_regression
mnist_predictor
toxicology_testing
.ruff_cache
screenshots
output.png U
Screenshot 2025-09-06 at 8.44.0... U
Screenshot 2025-09-06 at 8.49.4... U
Screenshot 2025-09-06 at 8.50.0... U
Screenshot 2025-09-06 at 8.50.2... U
Screenshot 2025-09-06 at 8.50.3... U
Screenshot 2025-09-06 at 8.57.0... U
Screenshot 2025-09-06 at 9.02.0... U
Makefile U
requirements.txt U
toxicology.ipynb U
toxicology.py 1, U
.gitignore
Makefile M
README.md
OUTLINE
TIMELINE
toxicology.py 1, U
toxicology.ipynb U
Screenshot 2025-09-06 at 9.02.09 PM.png U
output.png U
toxicology_testing > toxicology.ipynb > from sklearn.utils.class_weight import compute_class_weight
Generate
Code
Markdown
Interrupt
Restart
Clear All Outputs
Go To
Jupyter Variables
Outline
Python 3.10
Epoch 6/10
63/63 0s 5ms/step - accuracy: 0.9140 - loss: 0.3681 - val_accuracy: 0.9322 - val_loss: 0.2718
Epoch 7/10
63/63 0s 4ms/step - accuracy: 0.9241 - loss: 0.3371 - val_accuracy: 0.9309 - val_loss: 0.2595
Epoch 8/10
63/63 0s 4ms/step - accuracy: 0.9289 - loss: 0.3171 - val_accuracy: 0.9386 - val_loss: 0.2388
Epoch 9/10
63/63 0s 4ms/step - accuracy: 0.9362 - loss: 0.3028 - val_accuracy: 0.9322 - val_loss: 0.2479
Epoch 10/10
63/63 0s 4ms/step - accuracy: 0.9348 - loss: 0.2835 - val_accuracy: 0.9399 - val_loss: 0.2353
Validation Accuracy (Keras): 0.9399
25/25 0s 3ms/step
Validation Accuracy (sklearn): 0.9399
Validation F1-score: 0.3562
Validation ROC-AUC: 0.7661
train loss
val loss
Loss
Epoch


evaluation_accuracy_vs_iterations
0.9605
0.96
0.9595
0.959
700
750
800
850
900
950
1,000
1,050
1,100
1,150
1,200
1260
Run
Smoothed
Value
Step
Relative
validation
0.959
0.9591
1,260
2.893 sec