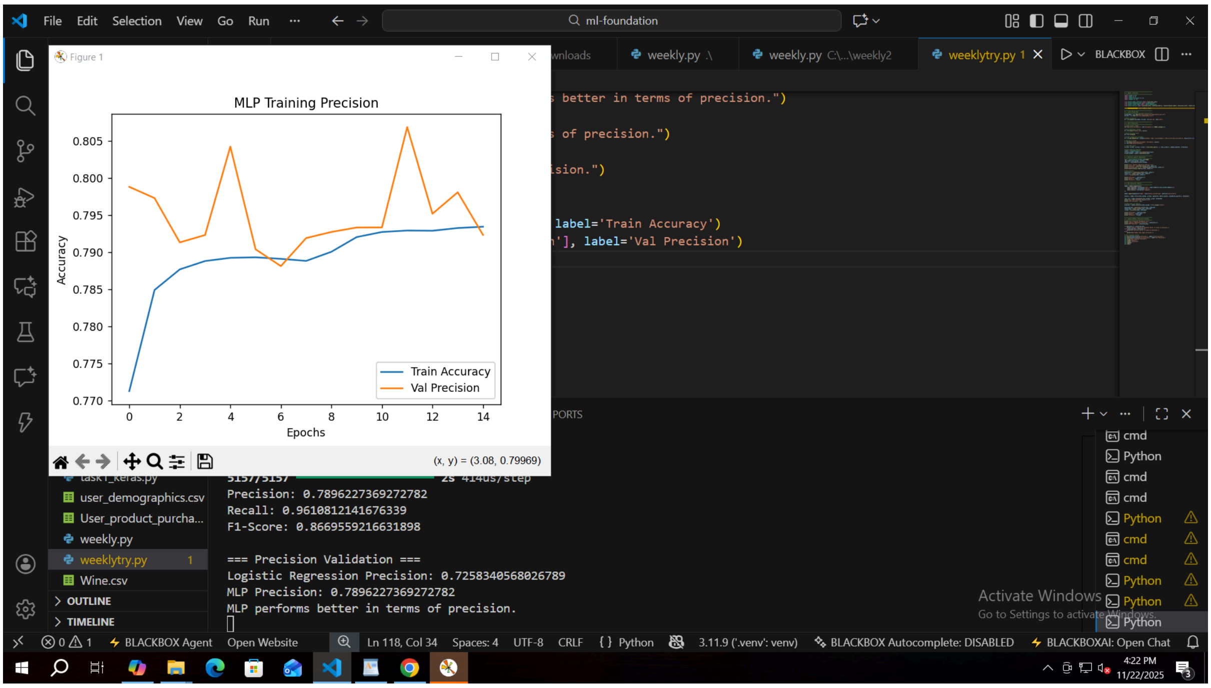
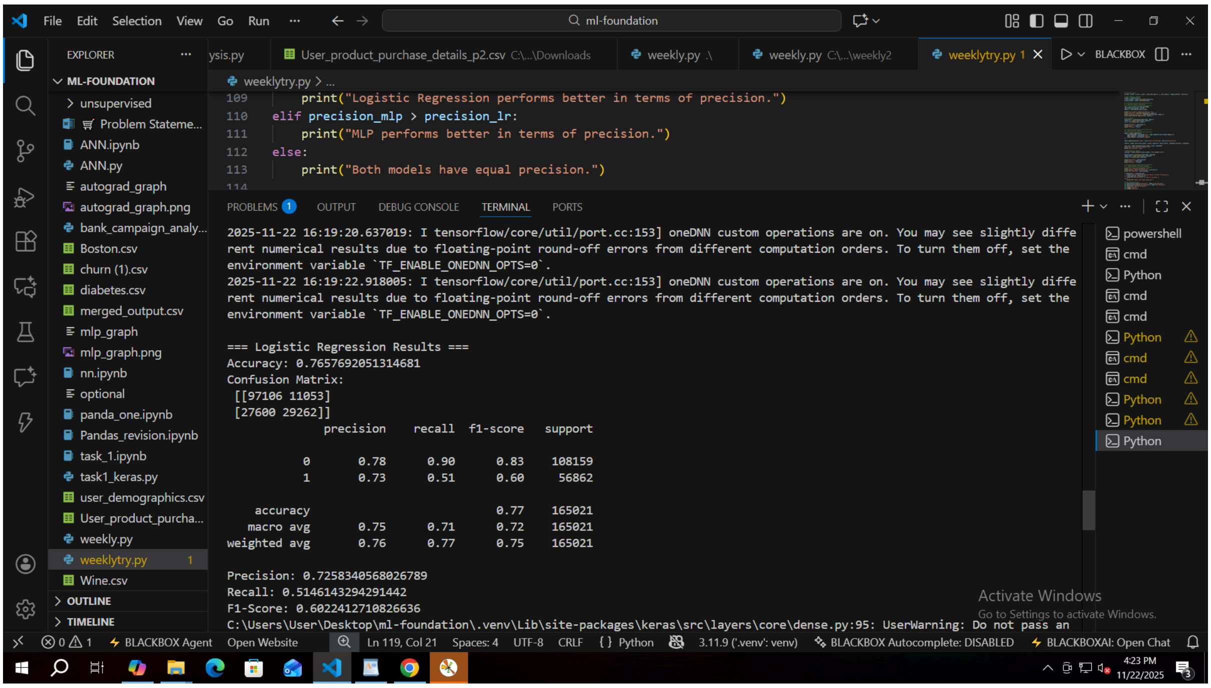


```
s better in terms of precision.")
 of precision.")
sion.")

label='Train Accuracy')
'], label='Val Precision')
```

```
5157/5157 ━━━━━━━━━━━━━━━━━━━━ 2s 414us/step
Precision: 0.7896227369272782
Recall: 0.9610812141676339
F1-Score: 0.8669559216631898

=== Precision Validation ===
Logistic Regression Precision: 0.7258340568026789
MLP Precision: 0.7896227369272782
MLP performs better in terms of precision.
```



```
    print("Logistic Regression performs better in terms of precision.")
elif precision_mlp > precision_lr:
    print("MLP performs better in terms of precision.")
else:
    print("Both models have equal precision.")
```

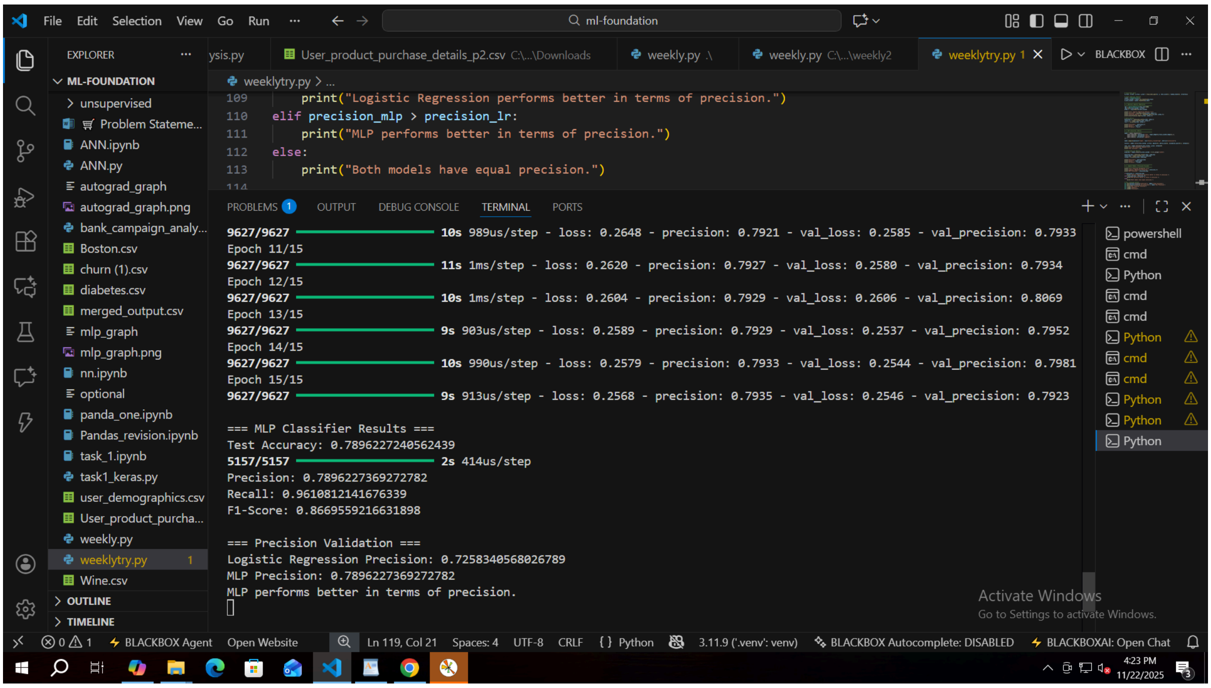
```
2025-11-22 16:19:20.637019: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly diffe
rent numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the
environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-11-22 16:19:22.918005: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly diffe
rent numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the
environment variable `TF_ENABLE_ONEDNN_OPTS=0`.

=== Logistic Regression Results ===
Accuracy: 0.7657692051314681
Confusion Matrix:
 [[97106 11053]
 [27600 29262]]
              precision    recall  f1-score   support

           0       0.78      0.90      0.83    108159
           1       0.73      0.51      0.60     56862

    accuracy                           0.77    165021
   macro avg       0.75      0.71      0.72    165021
weighted avg       0.76      0.77      0.75    165021

Precision: 0.7258340568026789
Recall: 0.5146143294291442
F1-Score: 0.6022412710826636
C:\Users\User\Desktop\ml-foundation\.venv\Lib\site-packages\keras\src\layers\core\dense.py:95: UserWarning: Do not pass an
```



```python
    print("Logistic Regression performs better in terms of precision.")
elif precision_mlp > precision_lr:
    print("MLP performs better in terms of precision.")
else:
    print("Both models have equal precision.")
```

```
9627/9627 ━━━━━━━━━━━━━━━━━━━━ 10s 989us/step - loss: 0.2648 - precision: 0.7921 - val_loss: 0.2585 - val_precision: 0.7933
Epoch 11/15
9627/9627 ━━━━━━━━━━━━━━━━━━━━ 11s 1ms/step - loss: 0.2620 - precision: 0.7927 - val_loss: 0.2580 - val_precision: 0.7934
Epoch 12/15
9627/9627 ━━━━━━━━━━━━━━━━━━━━ 10s 1ms/step - loss: 0.2604 - precision: 0.7929 - val_loss: 0.2606 - val_precision: 0.8069
Epoch 13/15
9627/9627 ━━━━━━━━━━━━━━━━━━━━ 9s 903us/step - loss: 0.2589 - precision: 0.7929 - val_loss: 0.2537 - val_precision: 0.7952
Epoch 14/15
9627/9627 ━━━━━━━━━━━━━━━━━━━━ 10s 990us/step - loss: 0.2579 - precision: 0.7933 - val_loss: 0.2544 - val_precision: 0.7981
Epoch 15/15
9627/9627 ━━━━━━━━━━━━━━━━━━━━ 9s 913us/step - loss: 0.2568 - precision: 0.7935 - val_loss: 0.2546 - val_precision: 0.7923

=== MLP Classifier Results ===
Test Accuracy: 0.7896227240562439
5157/5157 ━━━━━━━━━━━━━━━━━━━━ 2s 414us/step
Precision: 0.7896227369272782
Recall: 0.9610812141676339
F1-Score: 0.8669559216631898

=== Precision Validation ===
Logistic Regression Precision: 0.7258340568026789
MLP Precision: 0.7896227369272782
MLP performs better in terms of precision.
```