```python
acc = accuracy_score(y_test, pred_lr)
cm = confusion_matrix(y_test, pred_lr)
print(f"LR Accuracy: {acc:.4f}")
print("Confusion matrix:")
print(cm)
```

✓ 0.0s

```
LR Accuracy: 0.9849
Confusion matrix:
[[72163   177]
 [ 1479 36195]]
```

```
print(f"MLP Accuracy: {mlp_acc:.4f}")
print("Confusion matrix:")
print(mlp_cm)
```

✓ 0.0s

```
MLP Accuracy: 0.9966
Confusion matrix:
[[72006   334]
 [   38 37636]]
```

```
from sklearn.metrics import precision_score
lr_precision = precision_score(y_test, pred_lr)
mlp_precision = precision_score(y_test_arr, pred_mlp)
print(f"LR Precision: {lr_precision:.4f}")
print(f"MLP Precision: {mlp_precision:.4f}")
```

✓ 0.0s

```
LR Precision: 0.9951
MLP Precision: 0.9912
```

```python
#classification report
from sklearn.metrics import classification_report
print("Logistic Regression Classification Report:")
print(classification_report(y_test, pred_lr))
print("MLP Classification Report:")
print(classification_report(y_test_arr, pred_mlp))
```

[18] ✓ 0.0s

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

           0       0.98      1.00      0.99     72340
           1       1.00      0.96      0.98     37674

    accuracy                           0.98    110014
   macro avg       0.99      0.98      0.98    110014
weighted avg       0.99      0.98      0.98    110014

MLP Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     72340
           1       0.99      1.00      1.00     37674

    accuracy                           1.00    110014
   macro avg       1.00      1.00      1.00    110014
weighted avg       1.00      1.00      1.00    110014
```