

モデルの評価

前提知識

精度、再現率、F1スコア、混同行列

精度・再現率・F1スコアの計算

```
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

# Trainerの予測メソッド
predictions = trainer.predict(eval_dataset)
y_true = predictions.label_ids
y_pred = predictions.predictions.argmax(axis=1)

# 精度・再現率・F1スコア
accuracy = accuracy_score(y_true, y_pred)
precision, recall, f1, _ = precision_recall_fscore_support(y_true, y_pred, average="binary")

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1 Score: {f1:.4f}")
```

混同行列の描写

```
import matplotlib.pyplot as plt
from sklearn.metrics import ConfusionMatrixDisplay,
confusion_matrix

# 混同行列
cm = confusion_matrix(y_true, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=["Negative", "Positive"])
disp.plot(cmap="Blues")
plt.title("Confusion Matrix")
plt.show()
```

学習曲線の描写

```
# ログから学習曲線を描画
```

```
import pandas as pd
```

```
logs = pd.DataFrame(trainer.state.log_history)
```

```
# 損失関数 (loss) の推移
```

```
plt.plot(logs["epoch"], logs["loss"], label="Training Loss")
```

```
if "eval_loss" in logs.columns:
```

```
    plt.plot(logs["epoch"], logs["eval_loss"], label="Validation Loss")
```

```
plt.xlabel("Epoch")
```

```
plt.ylabel("Loss")
```

```
plt.title("Loss Curve")
```

```
plt.legend()
```

```
plt.show()
```