Sentiment Classification with Naive Bayes & Logistic Regression

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

# 1. Verisetin
reviews = [
    "This movie was amazing!",
    "Not bad, but not great either.",
    "Worst movie I've ever seen.",
    "Absolutely loved the story!",
    "I didn't like the acting.",
    "It was okay, nothing special.",
    "Totally blown away by the visuals.",
    "Terrible plot and weak script."
]

sentiments = [
    "Positive",
    "Neutral",
    "Negative",
    "Positive",
    "Negative",
    "Neutral",
    "Positive",
    "Negative"
]

# 2. TF-IDF ile vektörleştirme
vectorizer = TfidfVectorizer(stop_words='english')
X = vectorizer.fit_transform(reviews)

# 3. Train-test bölme
X_train, X_test, y_train, y_test = train_test_split(X, sentiments, test_size=0.25, random_state=42)

# 4. Naive Bayes modeli
nb_model = MultinomialNB()
nb_model.fit(X_train, y_train)
nb_preds = nb_model.predict(X_test)

# 5. Logistic Regression modeli
lr_model = LogisticRegression(max_iter=200)
lr_model.fit(X_train, y_train)
lr_preds = lr_model.predict(X_test)

# 6. Naive Bayes sonuçları
print("🔵 Naive Bayes Results:")
print("Accuracy:", accuracy_score(y_test, nb_preds))
print(classification_report(y_test, nb_preds))

# 7. Logistic Regression sonuçları
print("\n🟢 Logistic Regression Results:")
print("Accuracy:", accuracy_score(y_test, lr_preds))
print(classification_report(y_test, lr_preds))

# 8. 🎯 Yeni yorumlarla tahmin
new_reviews = [
    "I hated the entire movie.",
    "It was fantastic, I loved every minute!",
    "Meh, it was just okay I guess."
]

new_vectors = vectorizer.transform(new_reviews)

nb_predictions = nb_model.predict(new_vectors)
lr_predictions = lr_model.predict(new_vectors)

print("\n🔮 New Predictions:\n")
for i, review in enumerate(new_reviews):
    print(f"Review: {review}")
    print(f"Naive Bayes Prediction: {nb_predictions[i]}")
    print(f"Logistic Regression Prediction: {lr_predictions[i]}")
    print("-" * 50)
```

```
🔵 Naive Bayes Results:
Accuracy: 0.0
              precision    recall  f1-score   support

     Neutral       0.00      0.00      0.00       2.0
    Positive       0.00      0.00      0.00       0.0

    accuracy                           0.00       2.0
   macro avg       0.00      0.00      0.00       2.0
weighted avg       0.00      0.00      0.00       2.0


🟢 Logistic Regression Results:
Accuracy: 0.0
              precision    recall  f1-score   support

    Negative       0.00      0.00      0.00       0.0
     Neutral       0.00      0.00      0.00       2.0

    accuracy                           0.00       2.0
   macro avg       0.00      0.00      0.00       2.0
weighted avg       0.00      0.00      0.00       2.0


🔮 New Predictions:

Review: I hated the entire movie.
Naive Bayes Prediction: Positive
Logistic Regression Prediction: Positive
--------------------------------------------------
Review: It was fantastic, I loved every minute!
Naive Bayes Prediction: Positive
Logistic Regression Prediction: Positive
--------------------------------------------------
Review: Meh, it was just okay I guess.
Naive Bayes Prediction: Positive
Logistic Regression Prediction: Negative
--------------------------------------------------
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is i
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is i
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is i
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is i
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is i
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```