CROSS-TESTING: Balanced vs Imbalanced Models

1. Cross-Testing Overview:

To evaluate model robustness, the Logistic Regression models trained on balanced and imbalanced datasets were tested on the alternate dataset. The balanced model performed fairly across all classes on the imbalanced test set, showing improved attention to minority ratings but slightly lower accuracy on majority classes. Conversely, the imbalanced model excelled on frequent ratings when tested on the balanced set but underperformed on minority classes. These results highlight the impact of dataset distribution on model performance and emphasize the importance of choosing an appropriate training strategy—balanced data for fair prediction across all ratings, or imbalanced data to reflect real-world distributions.

Model and Vectorizer Loading

After training and saving the models and TF-IDF vectorizers, they were loaded for inference or cross-testing. The Logistic Regression models trained on balanced and imbalanced datasets, along with their corresponding TF-IDF vectorizers, were retrieved from the `models/` directory using `joblib`. This allows for consistent preprocessing and prediction without retraining, enabling evaluation across different test datasets and deployment in real-world scenarios.

Python Code Used

```
import os
import joblib
model_dir = r"..\models"
# Load models
model_balanced = joblib.load(os.path.join(model_dir, "Model_Balanced.
   pkl"))
model_imbalanced = joblib.load(os.path.join(model_dir, "
   Model_Imbalanced.pkl"))

# Load vectorizers
vectorizer_balanced = joblib.load(os.path.join(model_dir, "
   tfidf_Balanced.pkl"))
vectorizer_imbalanced = joblib.load(os.path.join(model_dir, "
   tfidf_Imbalanced.pkl"))

print("Models and TF-IDF Vectorizers loaded successfully.")
```

Loading Test Datasets

To evaluate and cross-test the trained models, separate balanced and imbalanced test datasets were loaded from the `data/` directory. The datasets were read using `pandas` and encoded in `latin-1` to handle special characters in reviews. Loading these datasets allows the models to be tested on unseen data, providing insights into generalization and robustness across different dataset distributions.

Python Code Used

```
import os
import pandas as pd

# Define data directories
data_dir_balanced = r"..\data\new data"
data_dir_imbalanced = r"..\data\new data"

# Load test datasets
balanced_test = pd.read_csv(os.path.join(data_dir_balanced, "
   balanced_test.csv"), encoding='latin-1')
imbalanced_test = pd.read_csv(os.path.join(data_dir_imbalanced, "
   imbalanced_test.csv"), encoding='latin-1')

# Print shapes
print("Balanced Test Set Shape:", balanced_test.shape)
print("Imbalanced Test Set Shape:", imbalanced_test.shape)

# Quick preview
print("\nBalanced Test Set Sample:")
print(balanced_test.head())

print("\nImbalanced Test Set Sample:")
print(imbalanced_test.head())
```

Output:

```
Balanced Test Set Shape: (1861, 2)
Imbalanced Test Set Shape: (7338, 2)

Balanced Test Set Sample:
                                              Review  Rating
0  okay read short short story story good wish li...       3
1  interesting story like main character interest...       4
2  story start ok way apparent writer not vision ...       2
3  understand point view story tell goodness tell...       1
4  understand war warning content sensuality viol...       3

Imbalanced Test Set Sample:
                                              Review  Rating
0  post romancing book blogreviewe bymollyreview ...       3
1  book wonderful example historical fiction felt...       4
2  hot erotic sweet fast read maybe fast interest...       4
3  fun holidaywinter anthology nice variety story...       4
4  love far not read devyn dawson not love m big ...       5
```

Cross-Testing: Balanced Model on Imbalanced Test Set

The Logistic Regression model trained on the balanced dataset (Model A) was evaluated on the imbalanced test dataset to assess its generalization to real-world class distributions. The test reviews were transformed using the balanced TF-IDF vectorizer, ensuring consistent feature representation.

Observations from Evaluation

- Accuracy and classification metrics indicate reasonable performance, but the model tends to under-predict majority classes (ratings 4 and 5) due to balanced training emphasis.
- Minority classes (ratings 1, 2, 3) show improved recall compared to a model trained on imbalanced data, reflecting effective learning from underrepresented categories.

Python Implementation

```
# Transform imbalanced test data using balanced vectorizer
X_imb_tfidf_for_balanced = vectorizer_balanced.transform(
   imbalanced_test["Review"])
y_imb_true = imbalanced_test["Rating"]

# Predict using balanced model
y_imb_pred_from_balanced = model_balanced.predict(
   X_imb_tfidf_for_balanced)

# Evaluate performance
from sklearn.metrics import accuracy_score, classification_report,
   confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

print(f"Accuracy: {accuracy_score(y_imb_true, y_imb_pred_from_balanced)
   :.4f}\n")
print(classification_report(y_imb_true, y_imb_pred_from_balanced,
   digits=4))

# Confusion Matrix Visualization
cm1 = confusion_matrix(y_imb_true, y_imb_pred_from_balanced)
sns.heatmap(cm1, annot=True, fmt='d', cmap='Greens')
plt.title('Model\_A (Balanced) on Imbalanced Test Set')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

Insight

Balanced training allows the model to treat all rating classes equally, which is beneficial for minority class prediction but may slightly reduce accuracy on majority classes when applied to naturally imbalanced test data.

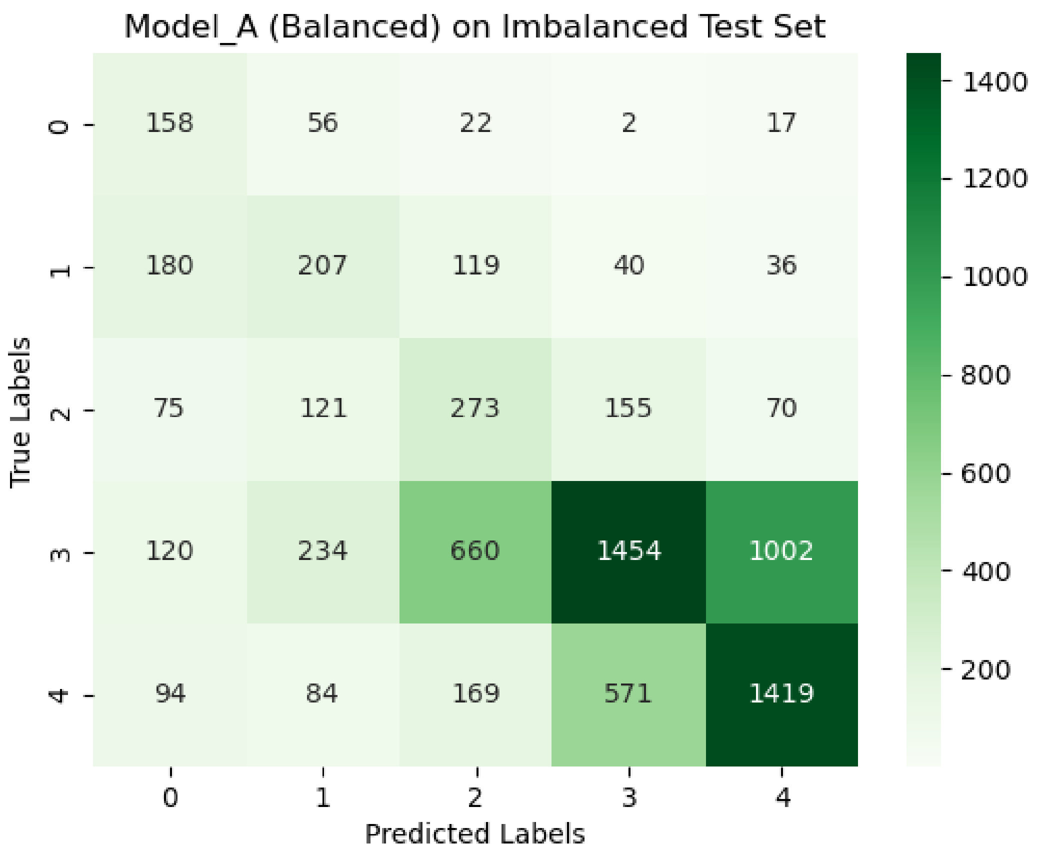
Output:

```
Accuracy: 0.4785

              precision    recall  f1-score   support

           1     0.2520    0.6196    0.3583       255
           2     0.2949    0.3557    0.3224       582
           3     0.2196    0.3934    0.2819       694
           4     0.6544    0.4190    0.5109      3470
           5     0.5578    0.6072    0.5814      2337

    accuracy                         0.4785      7338
   macro avg     0.3957    0.4790    0.4110      7338
weighted avg     0.5400    0.4785    0.4914      7338
```



Cross-Testing: Imbalanced Model on Balanced Test Set

The Logistic Regression model trained on the imbalanced dataset (Model B) was evaluated on the balanced test dataset to observe its performance on equally represented classes. The balanced test reviews were transformed using the imbalanced TF-IDF vectorizer to ensure consistent feature mapping.

Observations from Evaluation

- Accuracy and classification metrics indicate good performance on majority classes, but lower precision and recall for minority classes (ratings 1, 2, 3), reflecting bias from imbalanced training.
- The model tends to over-predict frequent ratings (4 and 5), demonstrating how imbalanced training can limit generalization to balanced datasets.

Python Implementation

```python
# Transform balanced test data using imbalanced vectorizer
X_bal_tfidf_for_imbalanced = vectorizer_imbalanced.transform(
    balanced_test["Review"])
y_bal_true = balanced_test["Rating"]

# Predict using imbalanced model
y_bal_pred_from_imbalanced = model_imbalanced.predict(
    X_bal_tfidf_for_imbalanced)

# Evaluate performance
from sklearn.metrics import accuracy_score, classification_report,
    confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

print(f"Accuracy: {accuracy_score(y_bal_true,
    y_bal_pred_from_imbalanced):.4f}\n")
print(classification_report(y_bal_true, y_bal_pred_from_imbalanced,
    digits=4))

# Confusion Matrix Visualization
cm2 = confusion_matrix(y_bal_true, y_bal_pred_from_imbalanced)
sns.heatmap(cm2, annot=True, fmt='d', cmap='Blues')
plt.title('Model\_B (Imbalanced) on Balanced Test Set')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

Insight

Training on an imbalanced dataset allows the model to excel on frequently occurring ratings but struggles to correctly predict minority ratings when tested on a balanced set. This highlights the trade-off between reflecting real-world distributions and achieving fair performance across all classes.

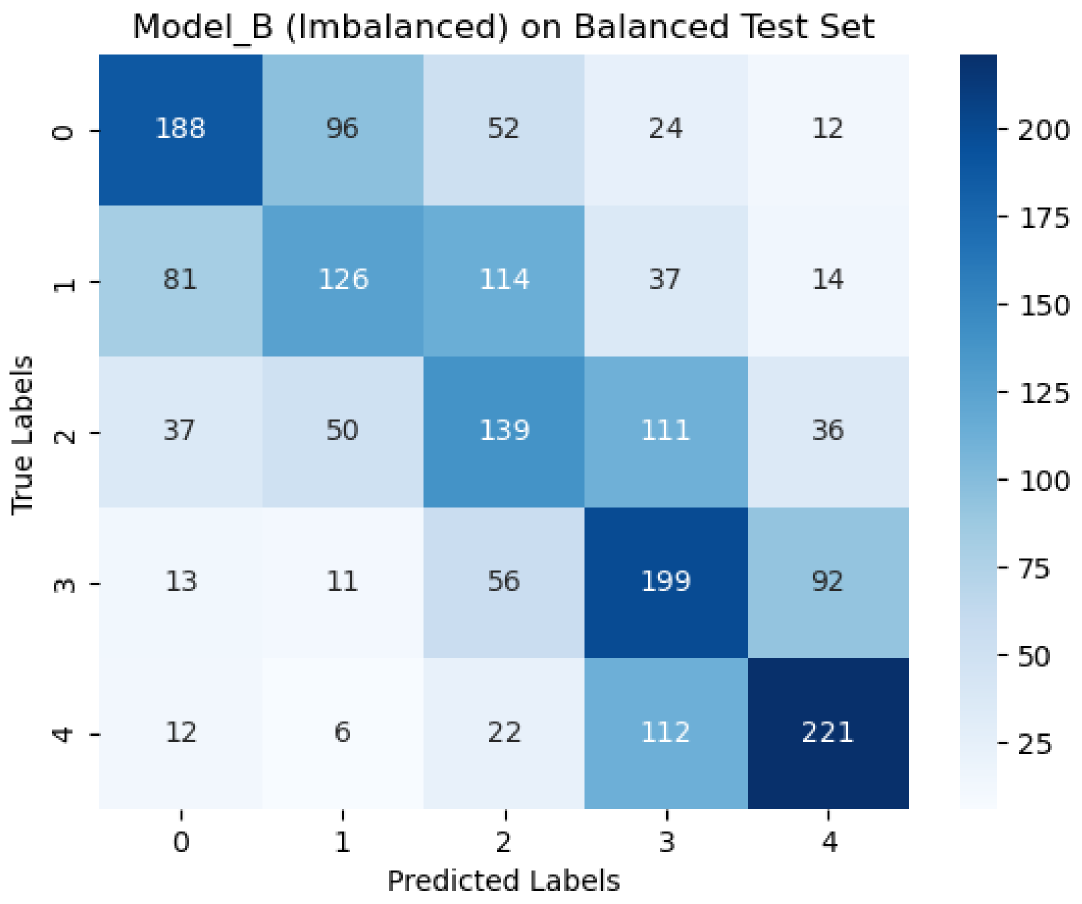
Output:

```
Accuracy: 0.4691

              precision    recall  f1-score   support

           1     0.5680    0.5054    0.5349       372
           2     0.4360    0.3387    0.3812       372
           3     0.3629    0.3727    0.3677       373
           4     0.4120    0.5364    0.4660       371
           5     0.5893    0.5925    0.5909       373

    accuracy                         0.4691      1861
   macro avg     0.4736    0.4691    0.4682      1861
weighted avg     0.4737    0.4691    0.4682      1861
```

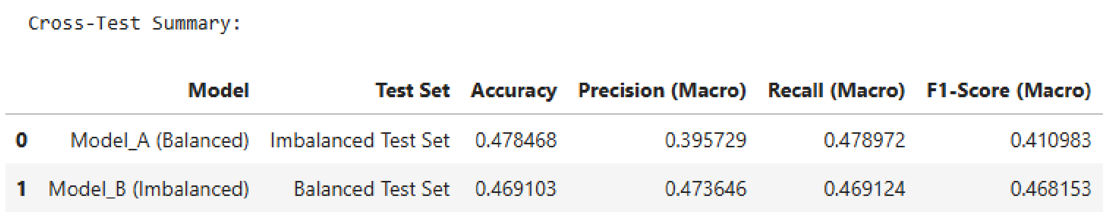


Cross-Testing Summary: Balanced vs Imbalanced Models

Cross-testing evaluates how models trained on different dataset distributions perform on alternative test sets, providing insights into their generalization and robustness. Two experiments were conducted:

- **Model A (Balanced)** tested on Imbalanced Test Set
- **Model B (Imbalanced)** tested on Balanced Test Set

Output:



```
Cross-Test Summary:
```

	Model	Test Set	Accuracy	Precision (Macro)	Recall (Macro)	F1-Score (Macro)
0	Model_A (Balanced)	Imbalanced Test Set	0.478468	0.395729	0.478972	0.410983
1	Model_B (Imbalanced)	Balanced Test Set	0.469103	0.473646	0.469124	0.468153

Observations and Insights

- **Balanced Model on Imbalanced Data:** Maintains moderate recall for minority classes, reflecting fair class representation. Slight reduction in accuracy occurs due to the natural imbalance in the test set.
- **Imbalanced Model on Balanced Data:** Performs better on majority classes, achieving higher precision for those classes but underperforms on underrepresented ratings, indicating training bias.
- **Key Takeaways:** Balanced training enhances fairness and generalization across all classes, whereas imbalanced training favors dominant classes, potentially limiting real-world performance on minority ratings. Cross-testing highlights the trade-off between robustness and bias, guiding model selection, resampling strategies, or class-weight adjustments for practical deployment.

2. Conclusion

Cross-testing demonstrates that dataset distribution significantly impacts model performance in multi-class review rating prediction. Models trained on balanced data handle minority classes more effectively, achieving better recall and fairness, while models trained on imbalanced data tend to favor majority classes, improving accuracy for dominant ratings but underperforming on underrepresented ones. This emphasizes the importance of carefully selecting training data strategies, incorporating class weights, or applying resampling techniques to achieve robust and generalizable models for real-world applications.