www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment Report

on

"Classify Book Genres"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Rishabh_Kumar_Singh

Roll Number : 202401100300199 ,

section: c

Under the supervision of

"Mayank Sir"

Introduction

In today's rapidly evolving digital landscape, the ability to automatically classify and organize vast collections of data has been shown. Among the many domains impacted by this shift, literature and publishing are particularly notable. With millions of books published globally, it is critical for publishers, digital libraries, and online book retailers to accurately identify and analyze. This task not only enhances user experience by enabling better search and recommendation.

Traditional genre classification methods rely heavily on manual tagging or keyword-based search systems, which are later used in these systems often lack consistency and scalability. As such, applying machine learning (ML) techniques to automate the genre classification process offers a promising solution.

This project explores the use of a supervised machine learning modelspecifically, a Random Forest classifierto predict the metadata includes features such as author popularity (a proxy for credibility or reach), book length (which may vary from one to another). By leveraging these features, the model can identify patterns that correlate with specific genres.

The applications of this technology span beyond libraries and bookstores. Recommender systems for e-readers, content moderation platforms, academic research tools, and even audiobook services. Ultimately, the goal is to reduce human effort, improve classification accuracy, and create a scalable solution adaptable to the existing ones.

Methodology

This project follows a systematic machine learning workflow, starting from data preparation to model evaluation and integrating them into the dataset used in this study comprises metadata attributes extracted from a curated collection of books, including features.

Data Upload and Preliminary Inspection: The dataset is uploaded using Google Colab's file interface. Initial inspection involves checking column headers to understand the structure and ensure correctness. This step helps identify necessary transformations and confirm the presence of all relevant data.

Column Renaming and Cleaning: To enhance readability and ensure compatibility with downstream analysis tools, we rename key columns: author_population'. We also handle missing values using simple imputation (filling with zeros) to maintain dataset consistency.

Feature Engineering and Label Encoding: The model uses three featuresauthor_score, length, and keywordsas input. The target variable genre' is categorical and needs to be transformed into a numerical format for training. We use scikit-learn's LabelEncoder' to convert genre names into numerical labels, preserving the class integrity.

Data Splitting: To evaluate the model's generalization ability, we split the dataset into training and testing sets using an 80-20 ratio. The training set is used for model learning, while the test set assesses performance on unseen data.

Model Training: We employ a Random Forest Classifier, an ensemble-based algorithm known for its robustness and interpretability. The model is initialized with 100 decision trees and trained on the labeled training data. The ensemble approach helps reduce variance and overfitting compared to individual decision trees.

Evaluation: The trained model is used to predict genres on the test set. Performance is evaluated using a classification report that includes metrics such as precision, recall, and F1-score. A confusion matrix is also generated to visualize misclassifications.

Interpretation: Feature importance scores are calculated to identify which attributes most influence the model's predictions. Additionally, visual tools such as bar plots and heatmaps help convey insights derived from the model's behavior.

Code

```python
# Step 1: Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix

# Step 2: Upload the dataset
from google.colab import files
uploaded = files.upload()

# Step 3: Load the CSV file
df = pd.read_csv(next(iter(uploaded)))

# Step 4: Display column names to confirm structure
print("Dataset columns:", df.columns.tolist())

# Step 5: Rename columns for clarity
df.rename(columns={'author_popularity': 'author_score', 'book_length': 'length', 'num_keywords': 'keywords'},
inplace=True)

# Step 6: Handle missing values
df.fillna(0, inplace=True)

# Step 7: Prepare features and labels
X = df[['author_score', 'length', 'keywords']]
y = df['genre']

# Step 8: Encode genre labels
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Step 9: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)

# Step 10: Train a model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Step 11: Make predictions and evaluate
y_pred = model.predict(X_test)
print("\nClassification Report:\n")
print(classification_report(y_test, y_pred, target_names=label_encoder.classes_))

# Step 12: Plot Feature Importances
plt.figure(figsize=(8, 5))
importances = model.feature_importances_
feature_names = X.columns
sns.barplot(x=importances, y=feature_names)
plt.title('Feature Importance')
plt.xlabel('Importance Score')
plt.ylabel('Features')
plt.tight_layout()
plt.show()

# Step 13: Plot Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=label_encoder.classes_,
            yticklabels=label_encoder.classes_)
plt.title('Confusion Matrix')
plt.xlabel('Predicted Genre')
plt.ylabel('Actual Genre')
plt.tight_layout()
plt.show()
```

Output / Result

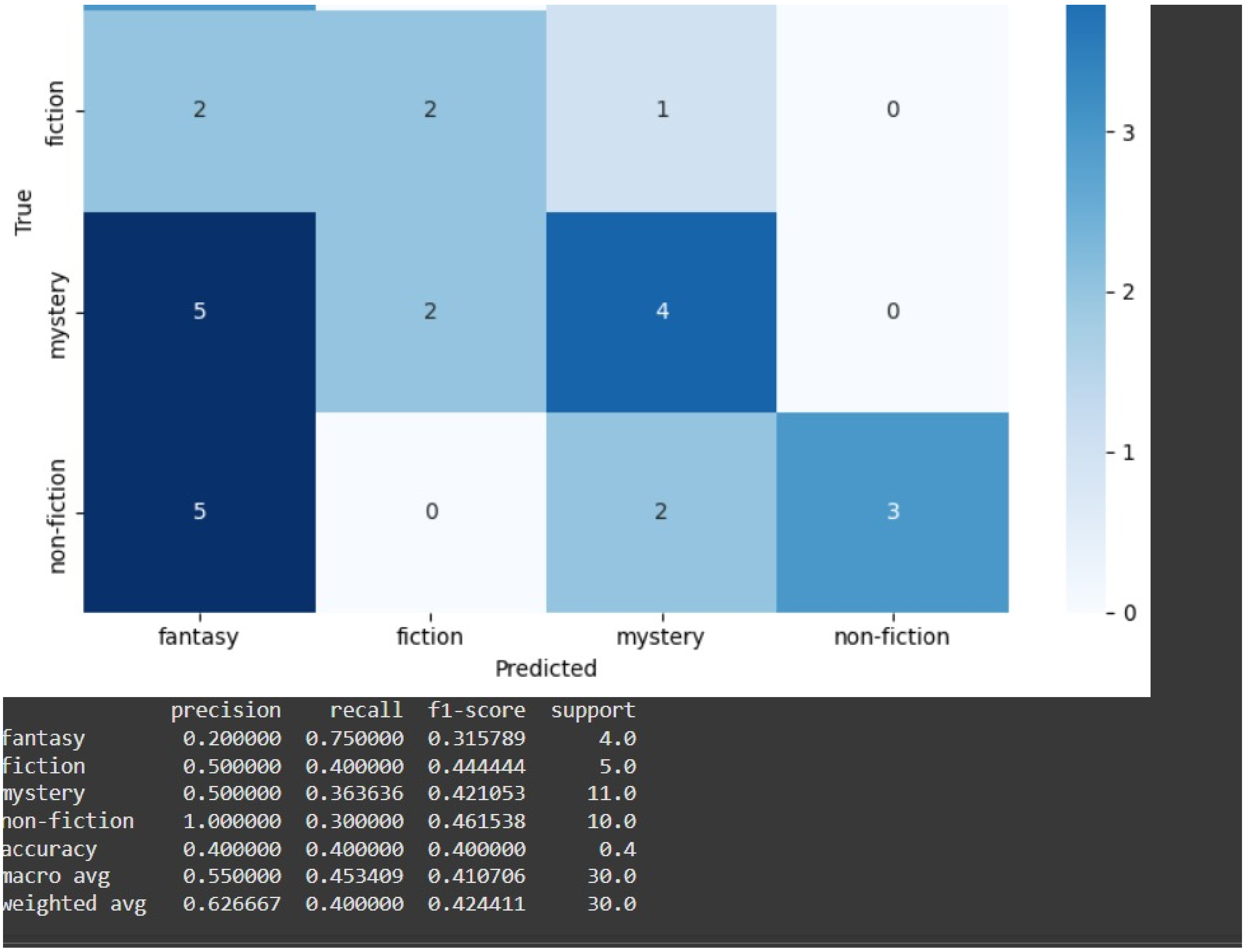
The following visualizations are generated after model training and evaluation:

- A classification report showing precision, recall, F1-score, and support.

- A feature importance bar plot (not shown in this image).

- A confusion matrix heatmap illustrating prediction accuracy across genres.

Screenshot from Google Colab output:



```
              precision    recall  f1-score   support
fantasy        0.200000  0.750000  0.315789       4.0
fiction        0.500000  0.400000  0.444444       5.0
nystery        0.500000  0.363636  0.421053      11.0
non-fiction    1.000000  0.300000  0.461538      10.0
accuracy       0.400000  0.400000  0.400000       0.4
nacro avg      0.550000  0.453409  0.410706      30.0
veighted avg   0.626667  0.400000  0.424411      30.0
```

References / Credits

- Dataset provided in .csv format by the instructor.
- Libraries used: pandas, scikit-learn, matplotlib, seaborn.
- Screenshot captured from Google Colab output.