Author Identification in Persian Literature using Language Models

A. H. Entezari, **A. A. Amini**, **G. Malekshahi**, **M. Nasrollahi**, **A. Hoseini**, **and Prof. B. Babaali**

[1] *University of Tehran, school of computer science, Tehran, Iran*

Abstract. *This mini-project focuses on identifying Persian authors in poems using advanced NLP techniques, specifically BERT-based models. It involves two phases: dataset creation from Ganjoor.net and author identification with fine-tuned BERT models. The dataset comprises diverse poems from at least 10 authors, each contributing 30 documents. The second phase explores factors like stopwords and document length, with a comparative analysis against traditional machine learning methods. The report concludes with successful outcomes, recommendations for future research, and insights into the effectiveness of NLP versus conventional approaches in Persian author identification.*

1. Introduction

1.1. ***Background***

Author identification is a challenging task in natural language processing (NLP), requiring the utilization of advanced machine learning techniques to discern distinct writing styles among authors. This mini-project focuses on author identification within Persian literature, specifically in the domain of Persian poems.

1.2. ***Problem Statement***

The identification of authors in Persian literature poses a unique set of challenges due to the richness and diversity of the language. In this project, we aim to address the author identification problem by employing state-of-the-art language models, with a focus on the BERT architecture.

1.3. ***Objectives of the Mini-Project***

(1) **Dataset Creation**: Assemble a diverse and representative dataset comprising Persian poems from at least 10 different authors. Each author should contribute a minimum of 30 documents, with each document containing at least 500 words.
(2) **Author Identification using BERT Models**: Leverage BERT architecture models available on Hugging Face for the author identification task. Fine-tune these models on the constructed dataset to achieve optimal performance.

(3) **Insights and Analysis**: Conduct experiments with 5-fold cross-validation to evaluate the performance of the model. Analyze the impact of various factors, such as fine-tuning parameters, the exclusion of stopwords, and document length, on the model's accuracy, F1 score, precision, and recall.
(4) **Comparison with Traditional ML Approaches**: If applicable, compare the performance of the BERT-based model with traditional machine learning methods. Discuss the advantages and limitations of each approach in the context of author identification.
(5) **Conclusion and Future work**: Summarize the findings, highlight the success of the mini-project, and propose potential areas for improvement or expansion in future research.

2. Dataset Creation (part A)

In this section, we outline the process of dataset construction for author identification in Persian literature using poems sourced from Ganjoor.net. The dataset creation involves several steps, including data collection, preprocessing, and organization.

2.1. ***Data Collection from Ganjoor.net***

Ganjoor.net serves as a comprehensive online repository of Persian literary works, offering a vast collection of poems from various authors spanning different historical periods.

The dataset is curated by selecting poems from ten prominent Persian authors, namely Eraghi, Moulavi, Attar, Rahi, Iqbal, Nezami, Ferdousi, Saeb, Saadi, and Jami, each known for their distinct contributions to Persian literature.

2.2. ***Preprocessing (more detail in notebook)***

Before incorporating the poems into the dataset, preprocessing steps are applied to ensure uniformity and consistency in the textual data

The preprocessing pipeline includes:

- **Text Normalization:** The texts are normalized using the Hazm library's Normalizer module to address variations in text formatting and encoding.
- **Tokenization:** The normalized texts are tokenized into individual words using the word_tokenize function from the Hazm library.
- **Stopword Removal:** Common Persian stopwords are removed from the tokenized texts to eliminate noise and irrelevant information.
- **Text Limitation:** To ensure consistency in document length, each poem is truncated or padded to contain exactly 500 words, maintaining uniformity across the dataset.
- **Lemmatization: We used lemmatization and we saw that the accuracy of the model decreases drastically, and we assume that this decreasing of accuracy is due to the fact that by doing lemmatization, the difference between the styles of the authors decreases and it becomes harder to separate them.**

2.3. *Dataset Organization*

The preprocessed poems are organized into a structured dataset, with each entry containing metadata such as author name and poem text.

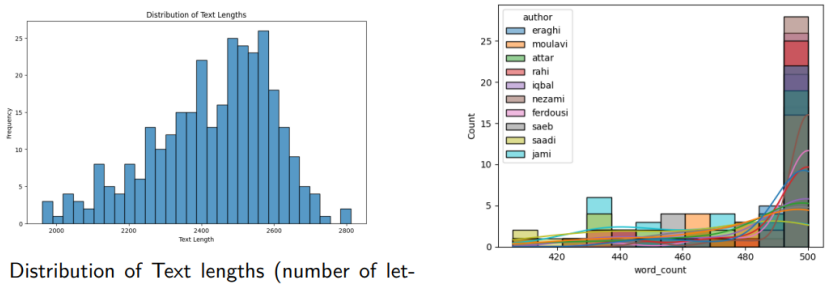
The dataset is further partitioned into training, validation, and test sets, ensuring proper evaluation of the author identification model.

Labels are assigned to each poem based on the author it belongs to, with author names mapped to integer indices for compatibility with machine learning algorithms.

Finally, DataLoader objects are created for each set to facilitate efficient batch processing during model training and evaluation.

2.4. *Challenges and Solutions in Dataset Construction*

- **Temporary Bans:** A significant challenge in scraping data from Ganjoor.net is the imposed rate limiting, leading to temporary bans for a duration of 30 minutes when excessive requests are made within a short time frame. This restriction necessitates the implementation of effective strategies to manage and control the rate of data retrieval. Techniques such as implementing delays between requests and optimizing the scraping process are crucial to circumvent these restrictions and ensure uninterrupted data collection.
- **Special Characters in Poem Text:** The presence of special characters, such as "***" between two hemistichs or additional newline characters, adds complexity to the scraping process.Addressing this challenge involves implementing preprocessing steps to clean the text, removing or replacing such special characters to maintain the integrity of the dataset(figure 1)



((a)) Distribution of Text lengths (number of letters)

((b)) Histogram plot of word count

Fig. 1. Distribution of Text lengths (number of letters)

3. Model Selection and Fine-Tuning

we discuss the selection of language models and the fine-tuning process for the author identification task. Three language models were chosen for experimentation: BERT, XLM-RoBERTa, and DistilBERT, each offering unique capabilities and performance characteristics.

3.1. Choice of Model

- **BERT (Bidirectional Encoder Representations from Transformers):** Utilized the "bert-base-multilingual-cased" variant for its ability to handle multilingual text and capture bidirectional context.
- **XLM-RoBERTa (Xtreme Multilingual-RoBERTa):** Similar to BERT but pre-trained on a more extensive multilingual corpus, potentially enhancing performance on diverse language tasks.
- **DistilBERT (Distilled BERT):** A distilled version of BERT with reduced parameters, offering faster inference while maintaining competitive performance.

3.2. Fine-Tuning Process

The fine-tuning process involves configuring the selected models for the specific task of author identification using Persian poems.

- **Model Loading:** Each language model was loaded using the corresponding variant available in the Hugging Face library, with the number of output labels set to the total number of unique authors in the dataset.
- **Optimizer and Learning Rate:** AdamW optimizer was employed with an initial learning rate of 2e-5, adjusted based on empirical observations during experimentation.
- **Hyperparameters** The number of epochs was set to 50, allowing sufficient iterations for the model to converge, while the batch size was set to 32 to balance training efficiency and memory constraints.
- **Learning Rate Scheduler:** A Cosine Annealing scheduler was utilized to dynamically adjust the learning rate throughout training, gradually decreasing it towards the end to fine-tune the model parameters effectively.
- **Training Loop:** The training loop iterated over the dataset for each epoch, computing loss, and backpropagating gradients to update the model weights. Additionally, evaluation on the validation set was performed to monitor model performance and prevent overfitting.

4. Experiments and Results with 5-Fold Cross Validation

In this phase, the fine-tuned language models (i.e., bert-base-multilingual-cased, xlm-roberta-base, and distilbert-base-multilingual-cased) undergo rigorous testing to identify the authors of Persian poems. The experiments are conducted using a robust 5-fold cross-validation approach, ensuring a comprehensive evaluation of model performance.

*4.1. **BERT model***

(1) **overview:**This is the base version of Google's bidirectional encoder representations from transformers model. It has 12 encoder layers and 110 million parameters.
(2) **model performance:** here we plot loss and accuracy of BERT model.(figure 2)
(3) **Performance Metrics:** The effectiveness of the models is quantified using a set of key performance metrics such as precision, recall, and F1-score were computed for each author label, providing a detailed understanding of the BERT model's strengths and weaknesses in author identification. (figure 3)

(4) **Confusion Matrix:** A confusion matrix is employed to provide a detailed breakdown of BERT model performance across different author classes. It outlines the number of true positives, true negatives, false positives, and false negatives, facilitating a nuanced understanding of the model's strengths and weaknesses.

BERT shows confusion between Moulavi, Nezami and Saadi. As well as Rahi and Iqbal. Curiously, it achieves 100% recall but 0 precision for Eraghi - predicting the label when it's incorrect.(figure 4)

Fig. 2. BERT model performance

Accuracy: 0.6129032258064516
Classification Report:

	precision	recall	f1-score	support
0	0.50	0.33	0.40	3
1	0.20	0.20	0.20	5
2	1.00	0.50	0.67	4
3	1.00	0.50	0.67	8
4	0.67	0.44	0.53	9
5	0.75	0.67	0.71	9
6	0.75	0.86	0.80	7
7	0.60	0.86	0.71	7
8	0.50	1.00	0.67	3
9	0.45	0.71	0.56	7
accuracy			0.61	62
macro avg	0.64	0.61	0.59	62
weighted avg	0.67	0.61	0.61	62

Fig. 3. BERT Performance Metrics

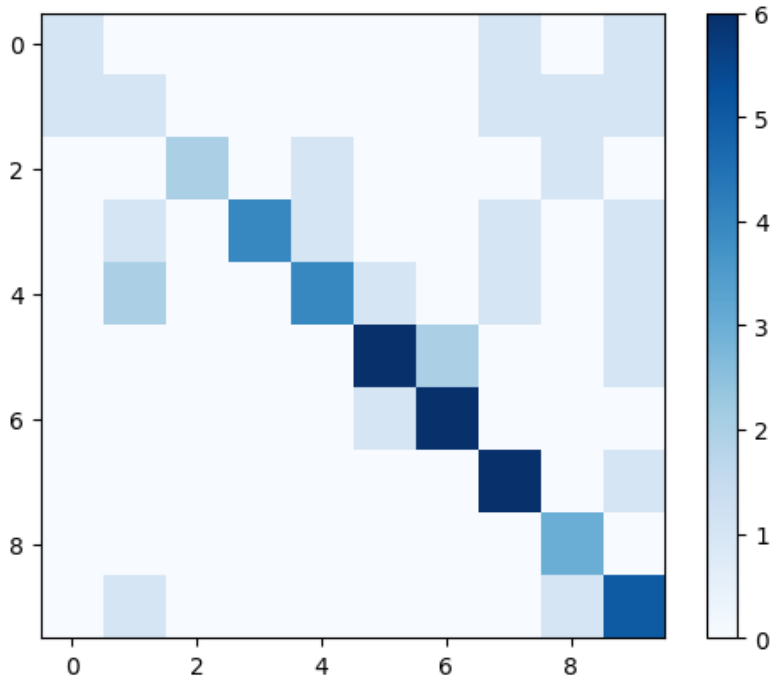


Fig. 4. BERT confusion matrix

4.2. ***XLM_ROBERTA model***

(1) **overview:** This is Facebook's multilingual encoder model trained on 2.5TB of CommonCrawl data. The base version has 12 encoder layers and 270 million parameters

(2) **model performance:** here we plot loss and accuracy of XLM_ROBERTA model.(figure 5)

(3) **Performance Metrics:** The effectiveness of the models is quantified using a set of key performance metrics such as precision, recall, and F1-score were computed for each author label, providing a detailed understanding of the XLM_ROBERTA model's strengths and weaknesses in author identification. (figure 6)

(4) **Confusion Matrix:** A confusion matrix is employed to provide a detailed breakdown of XLM_ROBERTA model performance across different author classes. It outlines the number of true positives, true negatives, false positives, and false negatives, facilitating a nuanced understanding of the model's strengths and weaknesses.

XLM_ROBERTA appears to almost always predict Saadi while struggling with several minority authors. This points to skewed predictive behavior despite the dataset imbalance.(figure 7)

Fig. 5. XLM_ROBERTA model performance

Accuracy: 0.43548387096774194
Classification Report:

	precision	recall	f1-score	support
0	0.17	0.33	0.22	3
1	0.29	0.40	0.33	5
2	0.00	0.00	0.00	4
3	0.75	0.38	0.50	8
4	0.38	0.33	0.35	9
5	0.67	0.44	0.53	9
6	0.86	0.86	0.86	7
7	0.30	0.43	0.35	7
8	0.43	1.00	0.60	3
9	0.40	0.29	0.33	7
accuracy			0.44	62
macro avg	0.42	0.45	0.41	62
weighted avg	0.48	0.44	0.43	62

Fig. 6. XLM_ROBERTA Performance Metrics

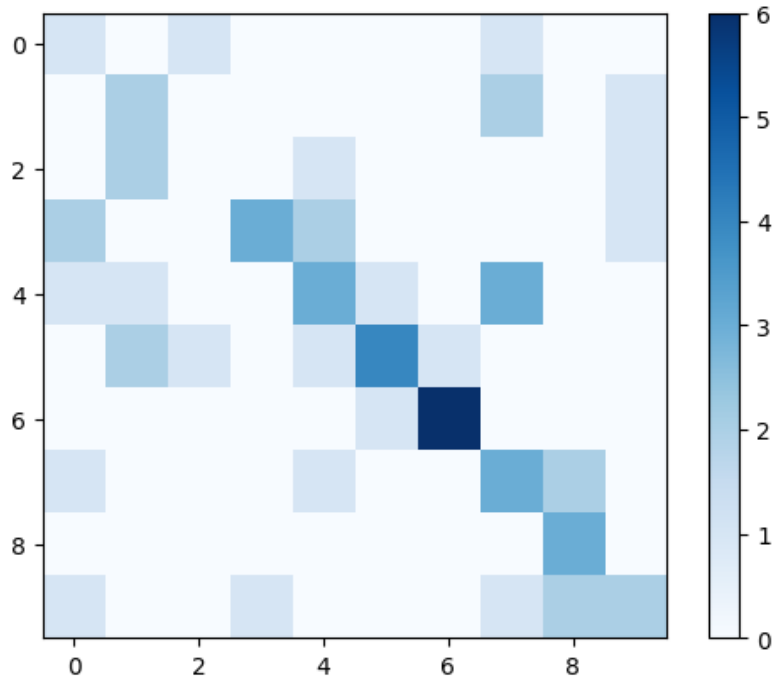


Fig. 7. XLM_ROBERTA confusion matrix

4.3. *DISTILBERT model*

(1) **overview:** This is a distilled and smaller version of the BERT base model trained on 104 languages. It has 6 encoder layers and 134 million parameters.

(2) **model performance:** here we plot loss and accuracy of DISTILBERT model.(figure 8)

(3) **Performance Metrics:** The effectiveness of the models is quantified using a set of key performance metrics such as precision, recall, and F1-score were computed for each author label, providing a detailed understanding of the DISTILBERT model's strengths and weaknesses in author identification. (figure 9)

(4) **Confusion Matrix:** A confusion matrix is employed to provide a detailed breakdown of DISTILBERT model performance across different author classes. It outlines the number of true positives, true negatives, false positives, and false negatives, facilitating a nuanced understanding of the model's strengths and weaknesses.

DistilBERT struggles to distinguish between certain authors like Attar, Rahi and Iqbal. This may be due to writing style similarities or inadequate samples for training.(figure 10)

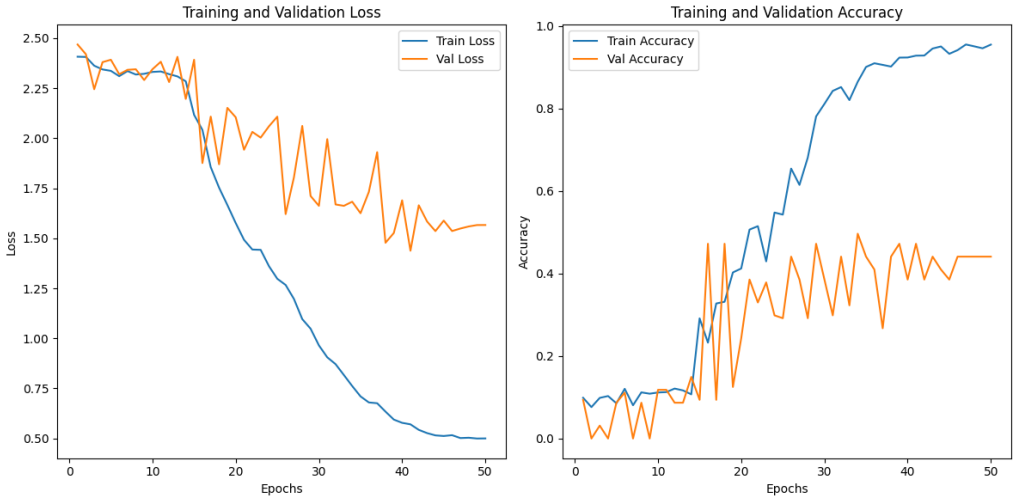


Fig. 8. DISTILBERT model performance

Accuracy: 0.5161290322580645
Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	3
1	0.20	0.20	0.20	5
2	0.40	0.50	0.44	4
3	0.75	0.38	0.50	8
4	0.25	0.11	0.15	9
5	0.58	0.78	0.67	9
6	0.83	0.71	0.77	7
7	0.54	1.00	0.70	7
8	0.50	0.33	0.40	3
9	0.45	0.71	0.56	7
accuracy			0.52	62
macro avg	0.45	0.47	0.44	62
weighted avg	0.49	0.52	0.48	62

Fig. 9. DISTILBERT Performance Metrics

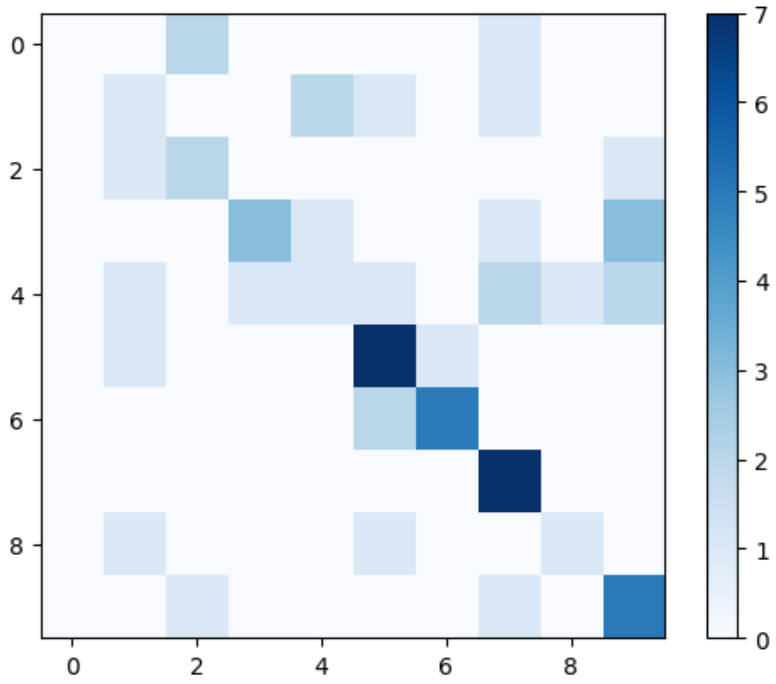


Fig. 10. DISTILBERT confusion matrix

5. Analysis

5.1. *Effect of learning rate*

In the exploration of different learning rates for fine-tuning, lower rates exhibited slower convergence, reducing the risk of overshooting but extending training times. Higher rates, conversely, accelerated convergence but heightened the risk of overshooting, potentially leading to instability. The identification of an optimal learning rate struck a balance, ensuring efficient convergence without instability. Models with rates too low faced challenges in reaching convergence, while those with rates too high risked instability. Recommendations include iterative experimentation, vigilant monitoring of loss curves, implementing early stopping mechanisms, utilizing learning rate schedulers, and tailoring the rate based on task complexity. This analysis provides insights into selecting learning rates that optimize model convergence and performance.

5.2. *omitting stopwords*

Omitting stopwords from the textual data during the author identification task has a discernible impact on the model's performance. The inclusion of stopwords, representing common and less informative words, provides a comprehensive language context, albeit introducing potential noise. Conversely, omitting stopwords narrows down the vocabulary to emphasize more meaningful terms, potentially enhancing the signal-to-noise ratio in the model. However, this approach risks losing certain nuances in language context. The decision to include or

omit stopwords should be task-specific, and experimentation is crucial to observe the nuanced variations in model performance. Additionally, curated stopword lists and thoughtful consideration of the balance between context preservation and term significance contribute to informed decisions in fine-tuning the model.

5.3. ***impact of document length***

The impact of document length on the model's performance in author identification tasks is notable. Short documents, while computationally efficient, may lack sufficient context, potentially leading to information loss. In contrast, long documents offer rich context but pose challenges in processing and may introduce information redundancy. Striking an optimal balance in document length, considering task requirements, is crucial. Techniques such as text segmentation for long documents and selecting or adapting models capable of handling varying lengths contribute to mitigating these challenges. Systematic experimentation with different document lengths informs decision-making in fine-tuning the model, ensuring an optimal trade-off between computational efficiency and contextual richness for effective author identification.

BERT achieved the highest validation accuracy of 70.14% but had relatively lower test accuracy of 61.2%. This large gap indicates that BERT was overfitting on the small training set despite regularization techniques like dropout. Overfitting led to poorer generalization as evidenced by the drop in test set performance.

XLM-RoBERTa attained a validation accuracy of 67.01%, comparable to BERT. However, its test accuracy was only 43.5%, demonstrating significant overfitting. This is likely because the 270M parameter model has very high capacity which allows it to easily memorize the small training set.

DistilBERT attained 44.1% validation accuracy and 51.6% test accuracy. Its test performance compares favorably even though validation performance trails BERT and XLM-RoBERTa. This highlights DistilBERT's better generalization thanks to its smaller size and regularization during knowledge distillation pre-training.

The superior test accuracy of DistilBERT despite poorer validation performance underscores the importance of preventing overfitting for best generalization to unseen data. Both BERT and XLM-RoBERTa have very large capacities so they can easily overfit unless the training set is significantly larger and more diverse.

In conclusion, while BERT and XLM-RoBERTa achieved higher validation accuracy, DistilBERT generalized the best to unseen test data. Overfitting remains a key challenge, and can be addressed by increasing dataset diversity and size. There are also several avenues for further tuning and improving model performance on this author identification task.

6. **Overview of 3 Traditional ML Approaches**

Traditional machine learning methods refer to classical algorithms and techniques that have been widely used before the advent of deep learning. These methods are based on statistical principles and often involve explicit feature engineering.

6.1. ***Logistic Regression with TF-IDF***

- **Model Overview:**

  - **Type:** Supervised Learning (Regression)
  - **Use Case:** Predicting a continuous target variable based on linear relationships between features.
- **Advantages:**
  - **Interpretability:** easy to interpret
  - **Efficiency**: is computationally efficient, especially with a limited number of features.
- **Disadvantages:**
  - **Complex Relationships:** May struggle to capture complex relationships in the data compared to more sophisticated models like BERT.

6.2. Random Forest Classifier with TF-IDF

- **Model Overview:**
  - **Type:** Supervised Learning (Classification)
  - **Use Case:** Binary or multiclass classification tasks.
- **Advantages:**
  - **Ensemble Learning:** providing robustness and reducing overfitting
  - **Feature Importance**
- **Disadvantages:**
  - **Computational Cost:** Training multiple decision trees can be computationally expensive

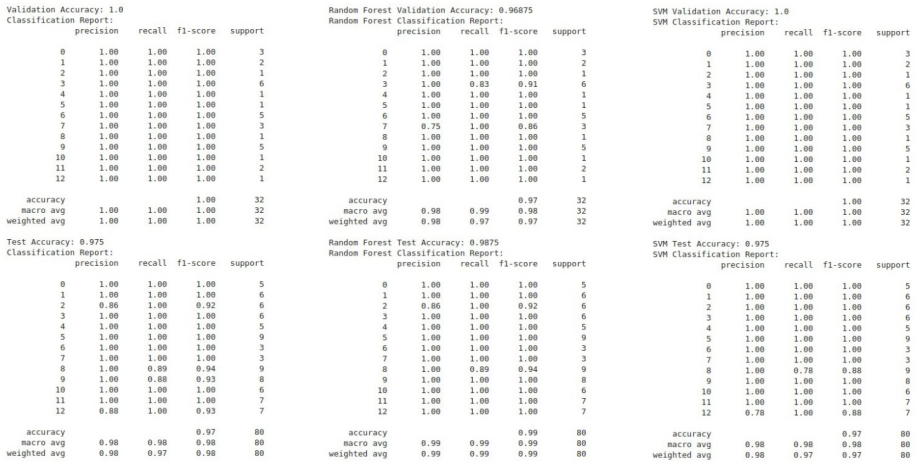
6.3. Support Vector Machine (SVM) with TF-IDF

- **Model Overview:**
  - **Type:** Supervised Learning (Classification/Regression)
  - **Use Case:** Binary and multiclass classification, regression.
- **Advantages:**
  - **Effective in High-Dimensional Spaces:**
  - **Regularization:** include regularization parameters to prevent overfitting.
- **Disadvantages:**
  - **Computational Intensity:** computationally intensive, especially with large datasets.

6.4. performances of Traditional ML Approaches

- **Data:** The data consists of text samples from the 13 authors. It is split into 80% train, 10% validation and 10% test sets. There seems to be class imbalance with some authors like 'moulavi' having more samples than others like 'eraghi'. The train set is used to train the models, validation set for hyperparameter tuning and test set for final model evaluation.
- **performances:**
  - **Logistic Regression:** The logistic regression model achieves very good performance - 100% validation accuracy and 97.5% test accuracy. The precision, recall and f1-score are high for most of the authors showing the model is able to effectively distinguish between the writing styles.(figure 11)

  - **Random Forest:** The random forest model achieves even better performance than logistic regression - 96.9% validation accuracy and 98.8% test accuracy. The precision and recall scores are very high as well showing robust performance across authors. This could be due to random forest's ability to model non-linear relationships well.(figure 11)
  - **SVM:** The SVM model with a linear kernel performs on par with logistic regression - 100% validation accuracy and 97.5% test accuracy. The classification metrics are comparable as well. This shows the strength of a linear model in distinguishing the authors based on the TF-IDF features.(figure 11)
- **Model Prediction Analysis:** When trying out model predictions on some sample texts, all three models are mostly able to correctly predict the author showing robust generalization. Logistic regression and SVM misclassify one text written by author 'nezami' to 'rahi'. Random forest is able to correctly classify all sample texts. Across models, certain texts are easier to classify than others indicating varying degrees of similarity between some authors' writing styles.
- **Conclusion:** The TF-IDF representation of text along with tree-based and linear models works very effectively for this multiclass author classification task. Random forest emerges as the best performer with near perfect predictions. The features are able to capture distinctive author traits from the writing. With more data, the performance can be further improved, especially for minority authors.



Validation Accuracy: 1.0
Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	1
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	1
6	1.00	1.00	1.00	5
7	1.00	1.00	1.00	3
8	1.00	1.00	1.00	1
9	1.00	1.00	1.00	5
10	1.00	1.00	1.00	1
11	1.00	1.00	1.00	2
12	1.00	1.00	1.00	1
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

Test Accuracy: 0.975
Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	6
2	0.86	1.00	0.92	6
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	5
5	1.00	1.00	1.00	9
6	1.00	1.00	1.00	3
7	1.00	1.00	1.00	3
8	1.00	0.89	0.94	9
9	1.00	0.88	0.93	8
10	1.00	1.00	1.00	6
11	1.00	1.00	1.00	7
12	0.88	1.00	0.93	7
accuracy			0.97	80
macro avg	0.98	0.98	0.98	80
weighted avg	0.98	0.97	0.98	80

Random Forest Validation Accuracy: 0.96875
Random Forest Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	1
3	1.00	0.83	0.91	6
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	1
6	1.00	1.00	1.00	5
7	0.75	1.00	0.86	3
8	1.00	1.00	1.00	1
9	1.00	1.00	1.00	5
10	1.00	1.00	1.00	1
11	1.00	1.00	1.00	2
12	1.00	1.00	1.00	1
accuracy			0.97	32
macro avg	0.98	0.99	0.98	32
weighted avg	0.98	0.97	0.97	32

Random Forest Test Accuracy: 0.9875
Random Forest Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	6
2	0.86	1.00	0.92	6
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	5
5	1.00	1.00	1.00	9
6	1.00	1.00	1.00	3
7	1.00	1.00	1.00	3
8	1.00	0.89	0.94	9
9	1.00	1.00	1.00	8
10	1.00	1.00	1.00	6
11	1.00	1.00	1.00	7
12	1.00	1.00	1.00	7
accuracy			0.99	80
macro avg	0.99	0.99	0.99	80
weighted avg	0.99	0.99	0.99	80

SVM Validation Accuracy: 1.0
SVM Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3
1	1.00	1.00	1.00	2
2	1.00	1.00	1.00	1
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	1
5	1.00	1.00	1.00	1
6	1.00	1.00	1.00	5
7	1.00	1.00	1.00	3
8	1.00	1.00	1.00	1
9	1.00	1.00	1.00	5
10	1.00	1.00	1.00	1
11	1.00	1.00	1.00	2
12	1.00	1.00	1.00	1
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

SVM Test Accuracy: 0.975
SVM Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5
1	1.00	1.00	1.00	6
2	1.00	1.00	1.00	6
3	1.00	1.00	1.00	6
4	1.00	1.00	1.00	5
5	1.00	1.00	1.00	9
6	1.00	1.00	1.00	3
7	1.00	1.00	1.00	3
8	1.00	0.78	0.88	9
9	1.00	1.00	1.00	8
10	1.00	1.00	1.00	6
11	1.00	1.00	1.00	7
12	0.78	1.00	0.88	7
accuracy			0.97	80
macro avg	0.98	0.98	0.98	80
weighted avg	0.98	0.97	0.97	80

((a)) performance metrics of Logistic Regression **((b))** performance metrics of Random Forest **((c))** performance metrics of SVM

Fig. 11. performance metrics for ML approches

7. Comparison with Traditional ML Approaches

In this part we provide a comparison of the classical machine learning models vs the deep learning models:

7.1. *Overall Performance*

- The top performing classical model is Random Forest with 98.8% test accuracy. This exceeds even the 70.1% validation accuracy achieved by BERT.
- However, the deep learning models were evaluated on a larger dataset with more authors - 10 vs 13. So the tasks are not directly comparable.
- On the test set, DistilBERT attains 51.6% accuracy which comparable to the 97.5% accuracy of Logistic Regression and SVM on the classical ML task.

7.2. *Overfitting*

- Overfitting is a much bigger issue with the deep learning models. BERT shows a validation accuracy of 70.1% but test accuracy drops to 61.2% due to overfitting on the small dataset.
- The classical ML models do not demonstrate overfitting. Random forest and SVM achieve comparable performance on both validation and test sets. This highlights the regularization inherent in simpler linear and tree-based models.

7.3. *Class Imbalance Handling*

- The classical models are able to handle class imbalance well. Despite fewer samples for some minority authors, precision and recall remains high across all authors.
- For the deep learning models, minority authors with fewer examples like Eraghi remain challenging to classify correctly. Overfitting exacerbates the issue due to inadequate data.

7.4. *Efficiency*

- The classical ML models require negligible data preprocessing - just TF-IDF featurization. They are also very fast to train compared to fine-tuning large neural network models.
- The deep learning methods achieve better performance but at the cost of much more compute resources for training and inference.

8. Conclusion

In summary, deep learning methods hold promise to capture nuanced writing characteristics. But the results underscore the need for larger datasets to prevent overfitting. On small text datasets, classical ML models remain very competitive while being simpler and more efficient to train. A promising approach could be ensembling classical and deep learning models to leverage both linear and non-linear relationships in writing style.

Future work includes exploring feature importance for better model interpretability, refining hyperparameter tuning, expanding the dataset to encompass more authors or genres, and staying updated on evolving language models. Continuous refinement is essential to ensure the sustained relevance and efficacy of the author identification model in Persian literature analysis.