PROJECT TITLE: DETECTION OF AI-GENERATED ARABIC TEXT: A DATA MINING APPROACH

Taibah University

AMANI ABDULLAH ALSEHI

Course Instructor: Prof. Dr. Mohammed Al-Sarem

DECEMBER 14, 2025

Abstract

This research presents a model for recognizing AI-generated Arabic text verus humen-generated text. A set of linguistic features such as Number of multiple elongations, verbs , dual forms, Number of periods , and entity diversity, was developed , and BERT representations were used to extract semantic features. I developed and evaluated these multiple machine learning approaches for binary classification of Arabic text (human vs. AI-generated) using the KFUPM-JRCAI Arabic Generated Abstracts dataset (41,940 samples).

Multiple models, including logistic regression ,SVM, Random Forest , and a deep learning model, were trained on the BERT embedding (asafaya/bert-base-arabic). The dataset exhibited significant class imbalance (80% AI, 20% human) and was split into train (70%), validation (15%), and test (15%) sets with strict protocols to prevent data leakage.The result shoed that the deep learning model outperformed the traditional models with a achieved 90.18% F1-score, 98.34% accuracy, and 99.48% ROC-AUC on the test set on test set.

This study confirms the effectiveness of combining linguistic features and deep representation of detecting machine-generated texts in Arabic.

Introduction

This project aims to detect AI-generated Arabic text by exploiting linguistic characteristics unique to the Arabic language. The main objective is to design and evaluate effective features and models that can reliably distinguish between human-written and AI-generated content.

Related Work

Previous studies in the field of detecting automatically generated Arabic texts have explored several approaches that take into account the unique linguistic characteristics of the Arabic language. Some research has focused on traditional linguistic analysis by extracting surface-level features such as word and sentence length, vocabulary diversity, the use of punctuation marks, and the frequency of function words. These features have shown noticeable differences between human-written texts and machine-generated texts.

The studies indicate that combining manual linguistic features with deep representations represents a promising trend in the field of machine-generated text detection in Arabic, given the complexity of the language's morphological and stylistic structure.

Dataset Description

Dataset Source

The dataset used in this study is derived from the Arabic Generated Abstracts Dataset, publicly available on HuggingFace under the identifier: KFUPM-JRCAI/arabic-generated-abstracts

The dataset contains Arabic academic abstracts written by humans as well as abstracts generated by multiple large language models. It is designed to support research in Arabic natural language processing, particularly in tasks related to AI-generated text detection.

Dataset Construction

Each original human-written abstract is paired with several AI-generated versions produced by different language models. For the purpose of this study, the dataset was restructured into a binary classification format:

- **Label 0 (Human):** Original human-written abstracts
- **Label 1 (AI):** AI-generated abstracts

Duplicate texts were removed to avoid bias and data leakage.

Dataset Size

After preprocessing and restructuring, the final dataset consists of:

- **Total samples:** *41,940 texts*
- **Human-written texts:** *8,388 samples (20.0%)*
- **AI-generated texts:** *33,552 samples (80.0%)*

Data Splits

To prevent data leakage and ensure reliable evaluation, the dataset was split as follows:

- **Training set:** 70%
- **Validation set:** 15%
- **Test set:** 15%

Splitting was performed **before feature extraction**, and the test set was strictly isolated and never used during feature engineering, model training, or validation.

- Training Set (70%)

  Total: 25,578 samples

  Human: 5,872 (22.9%)

  AI: 19,706 (77.1%)

- Validation Set (15%)

  Total: 5,481 samples

Human: 1,258 (23.0%)

AI: 4,223 (77.0%)

- Test Set (15%)

  Total: 5,481 samples

  Human: 1,258 (23.0%)

  AI: 4,223 (77.0%)

Methodology

I start The project with first importing all dataset from Get-Hub , then I merge the split of AI with linked human text,I saved these all data set imported from git-hub named(dataset.csv) in data-> raw . In (dataset.csv) file I add new column to labeling as:

- Label=0 for text generated by AI
- Label=1 for Human -written text

This step was essential to supervised learning models , which is help model to recognize

The difference between humen and AI text, by connecting each text with correct label it has. Also, labeling in beginning in data_prepreation phase help to ensure safety of training and help to evaluate the models correctly with suring no integrated between text.

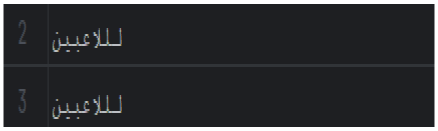
After labeling I split the file(dataset.csv) to 3 splits based on(70-15-15) (train-validation-test), I do this at beginning to ensure test set is never used for training or feature selection to avoid data leakage.After splitting and labeling all data set data ready to apply feature extraction as what I have depends on my number, my number is 6 my feature is (elongations, periods,verbs, dual words , and Entity Diversity)

Feature Engineering

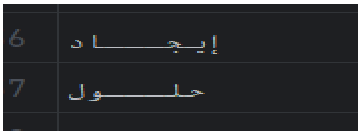
2.1 Number of elognations

In Detecting elongations I focused to identify words with repeated Arabic characters without misclassifying legitimate letter reptations. I used pattern with all Arabic character and pattern check if the word repeated more than 2 times .

Output: Its effective but some letter repeated 2 times but not included and other repeated more than 3 but the meaning in corrected not elongations .

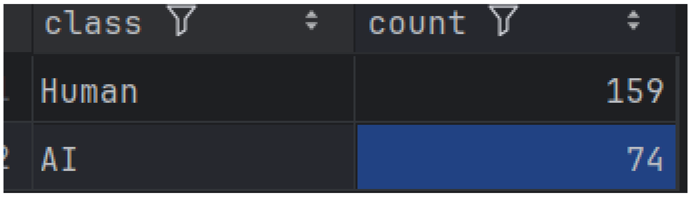


```
2  ـللاعبين
3  ـللاعبين
```



```
6  إيـــجـــــــا د
7  حـــلـــــــول
```

Why this feature important : Its helps to now the human use elongations more than AI , AI-text is in same pattern .

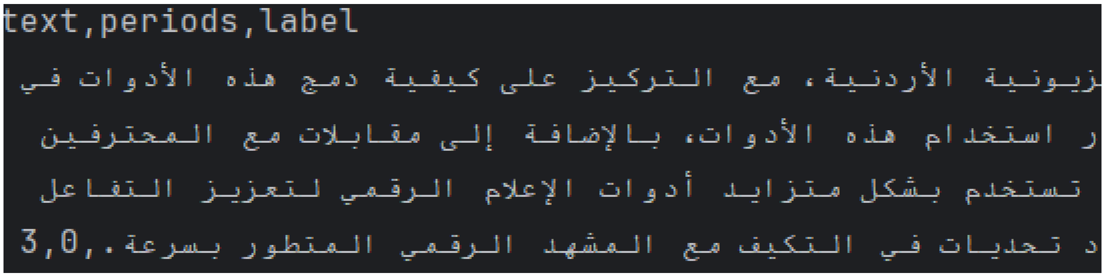


	class	count
1	Human	159
2	AI	74

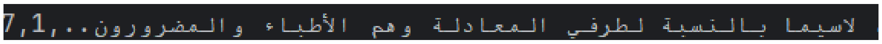
2.2:Number of Periods

In this feature I calculate Sentence Boundary . In Arabic the one dot (.) is used to end the sentence as same as in English . So , in this feature I take text and calculate how many (.) just one to ignore (…) which means etc…,

Output: Its effective and easy to calculate but in few sentence there is 2 dot and end its calculated as sentence boundary.

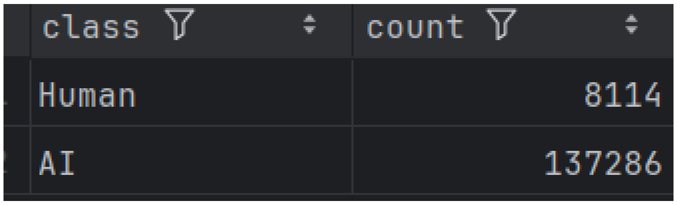


```
text,periods,label
زيونية الأردنية، مع التركيز على كيفية دمج هذه الأدوات في
ر استخدام هذه الأدوات، بالإضافة إلى مقابلات مع المحترفين
تستخدم بشكل متزايد أدوات الإعلام الرقمي لتعزيز التفاعل
د تحديات في التكيف مع المشهد الرقمي المتطور بسرعة.,3,0
```



```
لاسيما بالنسبة لطرفي المعادلة وهم الأطباء والمضرورون..,7,1
```

Why this feature important : In this feature I see clearly the AI used short sentence 17 times double human. Humen used dot 6% In the data set AI used dot 94.42% comparing with humen used. And Structured texts typically contain more periods also, AI may use periods differently than humans.



class	count
Human	8114
AI	137286

2.3 Number of verbs

First I want to mention that this part and next (dual verbs) talks about 50% of project times with multiple technical failures and implementation challenges .First , I implement rules based on stemming and tokenization techniques using NLTK evaluated. It lacks for Arabic and leads to a high number of false positives.

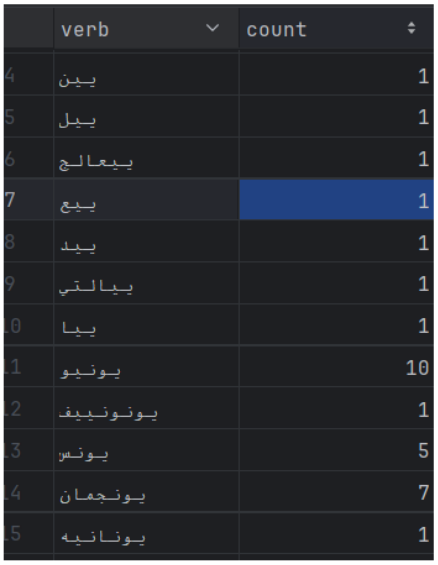
Second , the Farasa toolkit which its strong in Arabic NLP tasks. Farasa requires java-based component and linceses resources which mad it difficult to used in my python project.

Third, I try to use Camel tool but its not work first because I work on python version 10, I try to install helper library but still not work its work with python version 9 and older. So I rebuild all project in python v8.~ , and write the functions with camel tool but still not work .

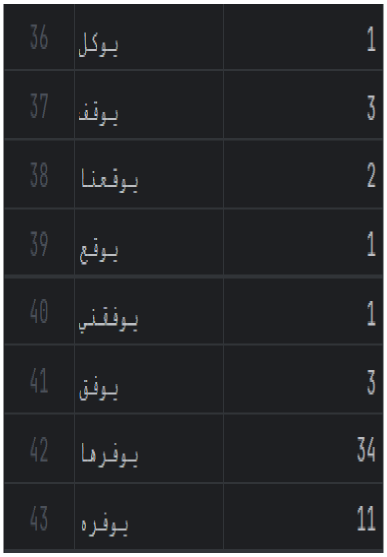
Finally , I implement verb heuristic rules and its approximate detection.To approximate verb usage without heavy analyzer .I applied prefix/suffix heuristic with exception handling , then a token is excludes from being a verb if it is too short, a definite noun, or in blacklist.

After that I normalize to reduce orthographic variance.

Output: this feature work bur its bring a lot of False positives .



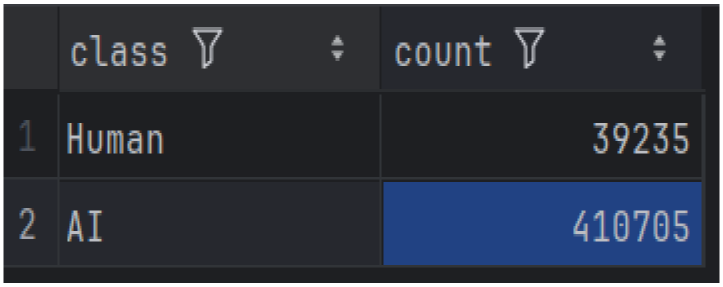
	verb	count
4	يين	1
5	ييل	1
6	ييعالج	1
7	ييع	1
8	ييد	1
9	ييالتي	1
10	ييا	1
11	يونيو	10
12	يونونييف	1
13	يونس	5
14	يونجمان	7
15	يونانيه	1



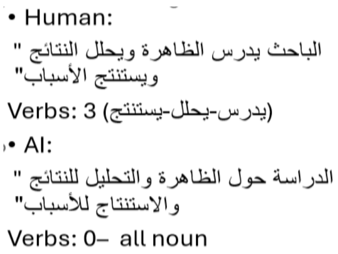
36	يوكل	1
37	يوقف	3
38	يوقعنا	2
39	يوقع	1
40	يوفقني	1
41	يوفق	3
42	يوفرها	34
43	يوفره	11

How this feature help?

- AI may use more uniform verb distribution
- Human writing shows greater verb variety
- Different verb tenses/forms between AI and human



	class	count
1	Human	39235
2	AI	410705



- Human:

" الباحث يدرس الظاهرة ويحلل النتائج ويستنتج الأسباب"

Verbs: 3 (يدرس-يحلل-يستنتج)

- AI:

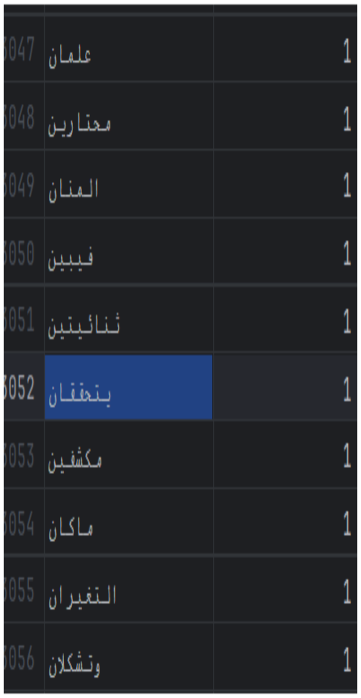
" الدراسة حول الظاهرة والتحليل للنتائج والاستنتاج للأسباب"

Verbs: 0– all noun

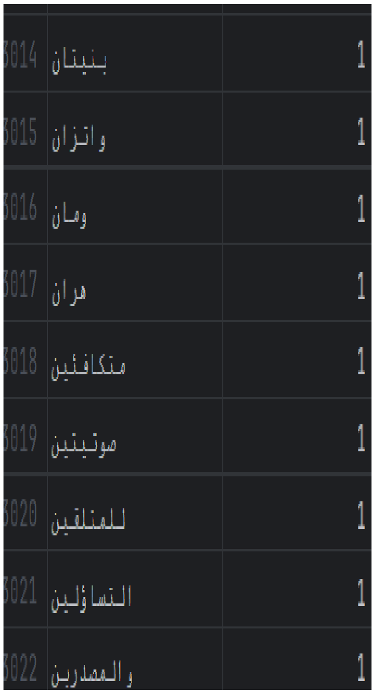
2.4: Number of dual words

In This feature like before first but blacklist words , because some words end with same prefix as dual ends. Then check the word length more than 2 letter and end with dual perffic .Also its heuristic way and bring many false positives.

Output:

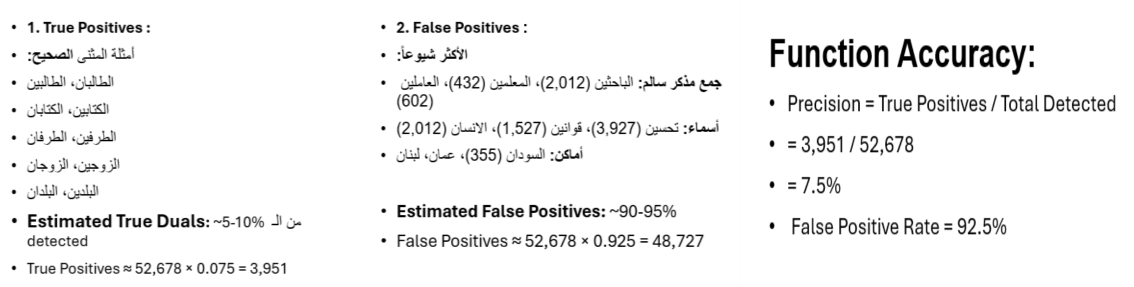


3047	علمان	1
3048	محتارين	1
3049	المنان	1
3050	فيبين	1
3051	ثنائيتين	1
3052	يتحققان	1
3053	مكشفين	1
3054	ماكان	1
3055	التغيران	1
3056	وتشكلان	1



3014	بنيتان	1
3015	واتزان	1
3016	ومان	1
3017	هران	1
3018	متكافئين	1
3019	موتيتين	1
3020	للمتلقين	1
3021	التساؤلين	1
3022	والمصدرين	1

Is this function works?



- **1. True Positives :**
- **أمثلة المثنى الصحيح:**
- الطالبان، الطالبين
- الكتابين، الكتابان
- الطرفين، الطرفان
- الزوجين، الزوجان
- البلدين، البلدان
- **Estimated True Duals:** ~5-10% من الـ detected
- True Positives ≈ 52,678 × 0.075 = 3,951

- **2. False Positives :**
- **الأكثر شيوعاً:**
- **جمع مذكر سالم:** الباحثين (2,012)، المعلمين (432)، العاملين (602)
- **أسماء:** تحسين (3,927)، قوانين (1,527)، الانسان (2,012)
- **أماكن:** السودان (355)، عمان، لبنان
- **Estimated False Positives:** ~90-95%
- False Positives ≈ 52,678 × 0.925 = 48,727

Function Accuracy:

- Precision = True Positives / Total Detected
- = 3,951 / 52,678
- = 7.5%
- False Positive Rate = 92.5%

How this feature help?

If we remove False Positives:

Assuming 7.5% are True Duals:

Human True Duals = 4,936 × 0.075 = 370 AI True Duals = 47,742 × 0.075 =3,581

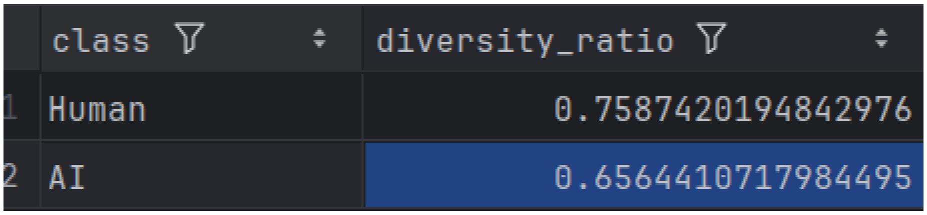
Ratio = 3,581 / 370 = 9.7:1

So, AI used dual 9times more than human.

2.5:Entity Diversity: Ration of unique entities to total entities

This feature detect how many important words in text. I take out non important work like connections , stop words, verbs, too short words . Then I divid the count number of words to total number of all words

Output: the out put shows that human text hase more entity diversity than AI-text.



	class	diversity_ratio
1	Human	0.7587420194842976
2	AI	0.6564410717984495

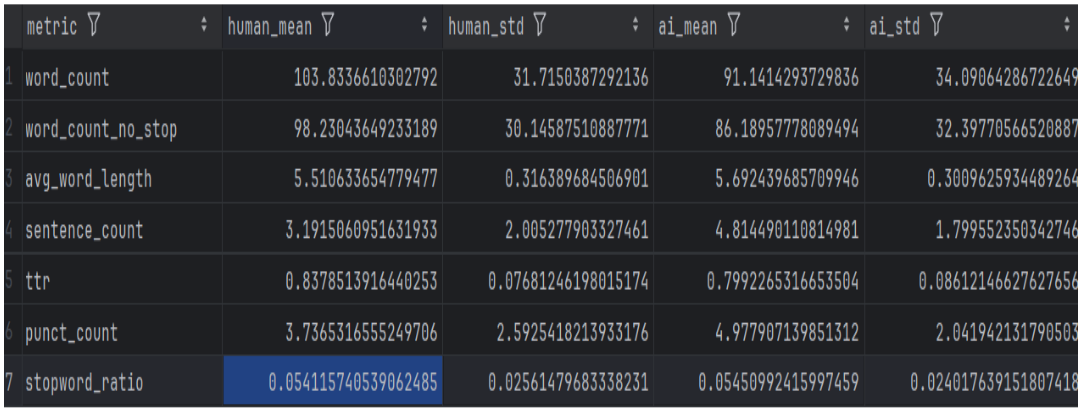
Also these 5 features including some Arabic-specific text preprocessing include:

- Normalization (e.g., converting to a standard, removing non-Arabic characters, handling hamza, alif maqsura).
- Removing diacritics (tashkeel).
- Stop word removal using a standard Arabic stop words list.
- Stemming was considered but not applied, as it may distort stylistic features essential for AI-text detection in Arabic.

Implementation of Exploratory Data Analysis (EDA)

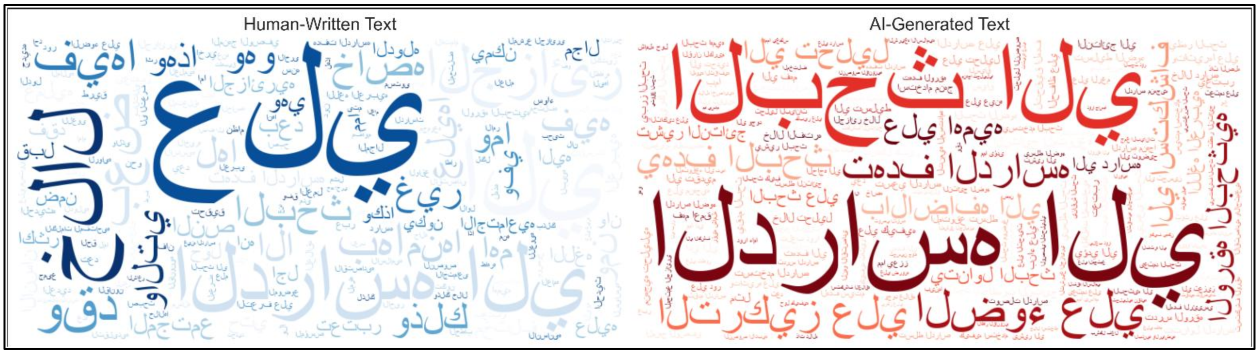
I implement EDA to find linguistic patterns and differences between the two classes. This include:

o Statistical Analysis: Average word length, sentence length, vocabulary richness (Type-Token Ratio).



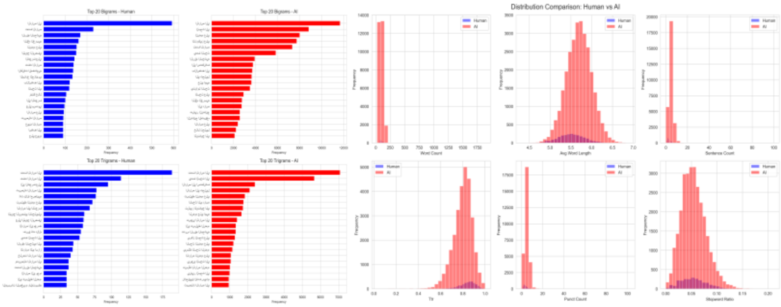
	metric	human_mean	human_std	ai_mean	ai_std
1	word_count	103.8336610302792	31.7150387292136	91.1414293729836	34.09064286722649
2	word_count_no_stop	98.23043649233189	30.14587510887771	86.18957778089494	32.39770566520887
3	avg_word_length	5.510633654779477	0.316389684506901	5.692439685709946	0.3009625934489264
4	sentence_count	3.1915060951631933	2.005277903327461	4.814490110814981	1.799552350342746
5	ttr	0.8378513916440253	0.07681246198015174	0.7992265316653504	0.08612146627627656
6	punct_count	3.7365316555249706	2.5925418213933176	4.977907139851312	2.041942131790503
7	stopword_ratio	0.054115740539062485	0.02561479683338231	0.05450992415997459	0.024017639151807418

o Visualization: Word clouds for each class, frequency distributions of top n-grams.



o Lexical Analysis: Comparing the use of function words, punctuation,

and specific terms. Also, these just for (training and validation split) until now test split not touched. The result of this function is



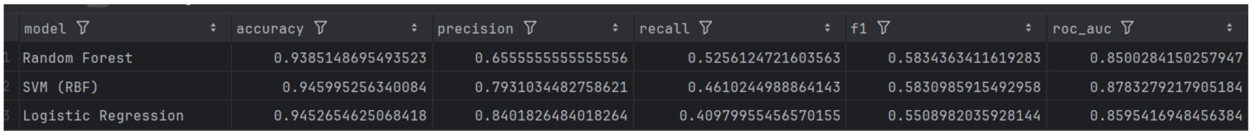
Modeling

In this phase I implement 4 models with training and evaluation to detect which one is effective , to use it in test split, and to comparing between traditional machine learning models and deep learning models.

First , preparing data to modeling so we split it and until now we just use training and evaluation splits. No used for test split until testing process. After that normalization feature scaling between 0-1 to feed first 3 models.

4.1 Traditional Machine Learning

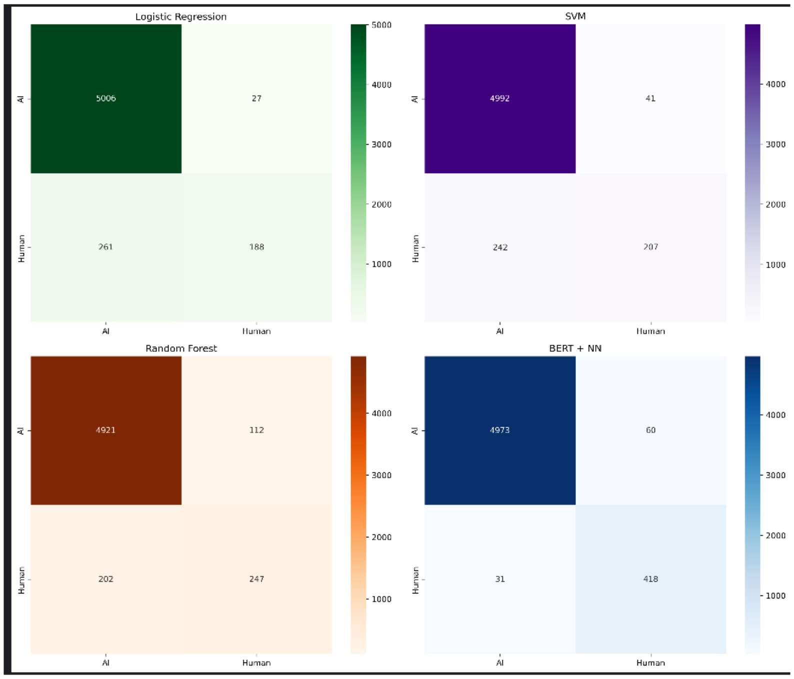
I train and evaluate 3 traditional models, **Logistic Regression** and its used as a baseline for othe models , and **Support Vector Machine (SVM)**, I used RBF Kernel was adopted in the SVM model to capture non-linear relationships between the extracted features, with probability estimation enabled to allow evaluation using the ROC-AUC metric. And I choose to apply **Random Forest** because its ability to capture non-linear relationships, feature importance analysis , and reducing overfitting.



	model	accuracy	precision	recall	f1	roc_auc
1	Random Forest	0.9385148695493523	0.6555555555555556	0.5256124721603563	0.5834363411619283	0.8500284150257947
2	SVM (RBF)	0.945995256340084	0.7931034482758621	0.4610244988864143	0.5830985915492958	0.8783279217905184
3	Logistic Regression	0.9452654625068418	0.8401826484018264	0.40979955456570155	0.5508982035928144	0.8595416948456384

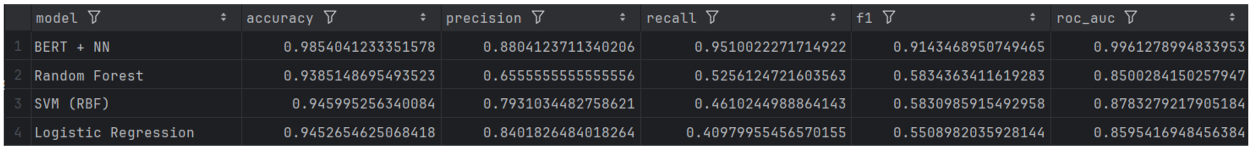
4.2 Deep Learning Model

I used (asafaya/bert-base-arabic) in BERT Embedding were the text streaming in BERT model as a batches for quality and to enhance memory usage , after that I train Feedforward Neural Netwo on train and validation splits.

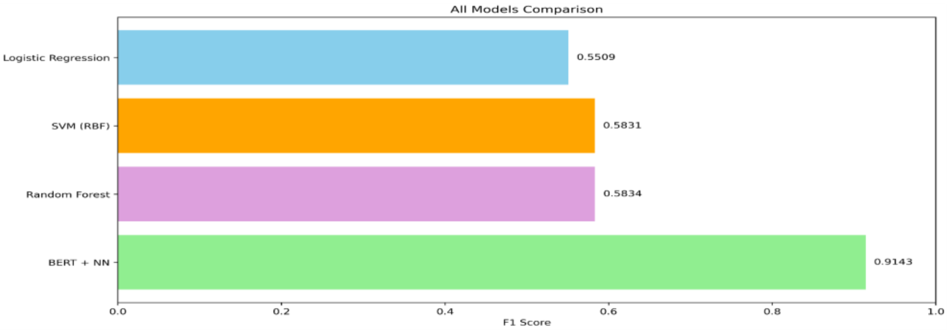


Results & Analysis

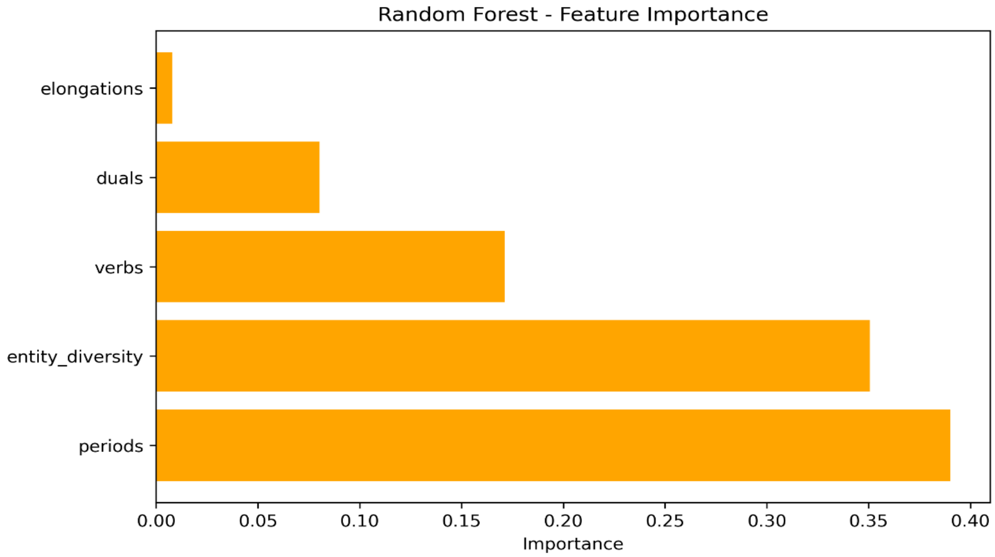
After training all 4 models I check the results depends on F1-score because its best measurements scale in unbalanced dataset . I saved all traing model result separately, and analytics figures, and confusion matrix to ensure reproducibility.



	model	accuracy	precision	recall	f1	roc_auc
1	BERT + NN	0.9854041233351578	0.8804123711340206	0.9510022271714922	0.9143468950749465	0.9961278994833953
2	Random Forest	0.9385148695493523	0.6555555555555556	0.5256124721603563	0.5834363411619283	0.8500284150257947
3	SVM (RBF)	0.945995256340084	0.7931034482758621	0.4610244988864143	0.5830985915492958	0.8783279217905184
4	Logistic Regression	0.9452654625068418	0.8401826484018264	0.40979955456570155	0.5508982035928144	0.8595416948456384



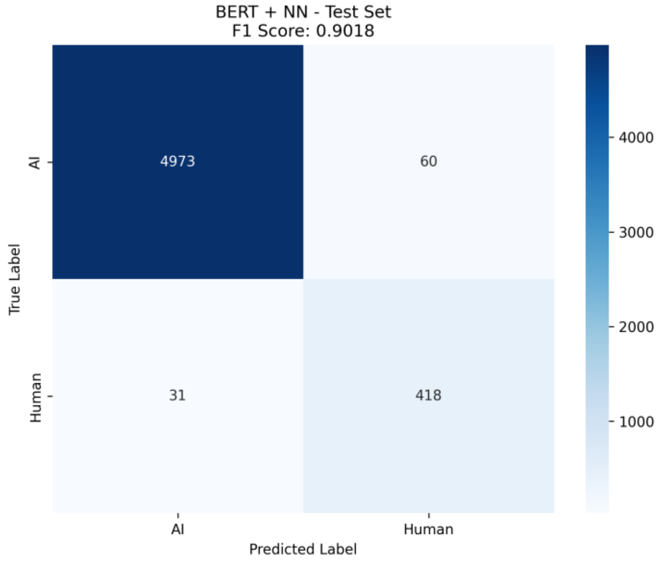
Feature visualization:



After training all models its clearly the BERT +NN is highest F1-score 0.91,

Which mean its more efficient than other model.

Then I used BERT on testing set , and here is the confusion matrix .



Conclusion & Future Work

This project addressed the task of detecting AI-generated Arabic text using both linguistic handcrafted features and deep learning models. Several Arabic-specific features were designed to capture stylistic and structural differences between human and AI-generated text. Traditional machine learning models achieved reasonable performance, but the BERT model significantly outperformed them, achieving an F1 score of 0.90 on the test set.

The results confirm that contextual embeddings from BERT are highly effective for modeling Arabic text and distinguishing AI-generated content from human writing.

Limitations

The project faced challenges related to the complex morphology of Arabic, reliance on heuristic rules for some features, and technical limitations with Arabic NLP tools. In addition, extracting BERT embeddings is computationally time expensive.

Future Work

Future work includes expanding the dataset, adopting more advanced Arabic language models, improving morphological analysis, and combining handcrafted features with deep embeddings to enhance performance and interpretability.