İlker Yayalar-201805066 – İrem Beyza Gül 201805073

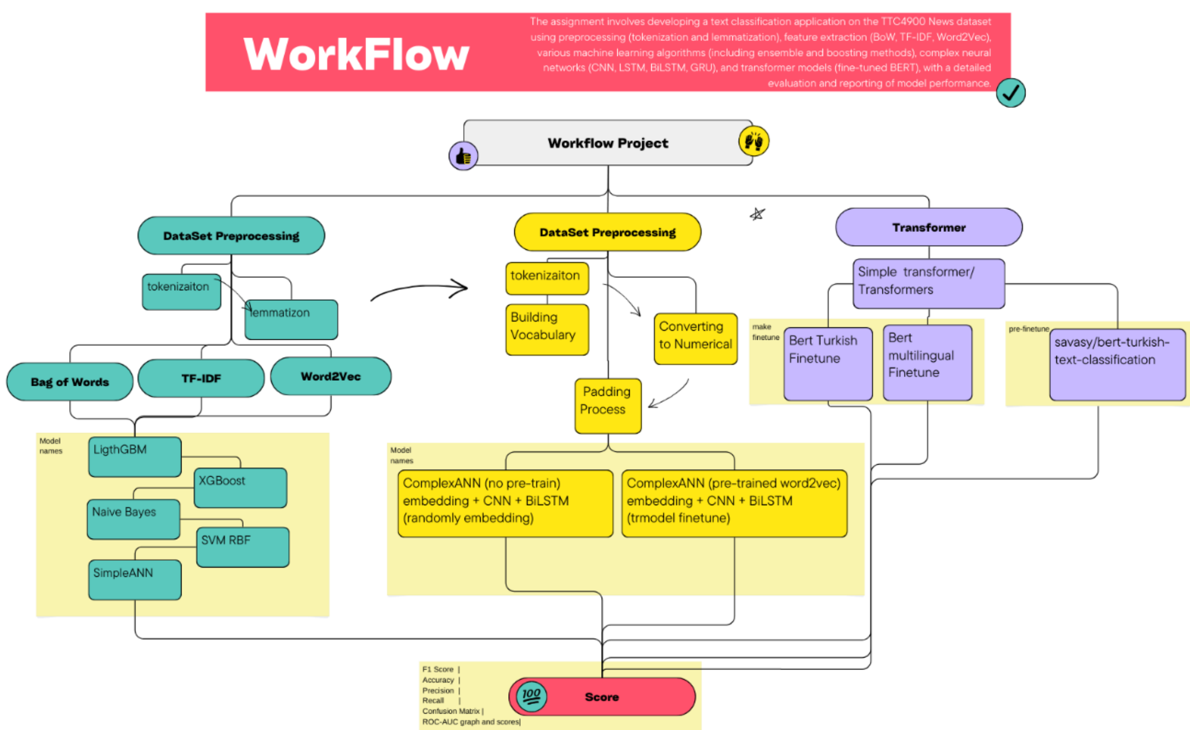
Project Objective

The goal of this project is to classify texts from the TTC4900 News Dataset, which consists of **4,900 samples**. The workflow involves data preprocessing (tokenization and lemmatization), feature extraction (BoW, TF-IDF, Word2Vec), various machine learning algorithms, complex neural networks (CNN, LSTM, BiLSTM, GRU), and Transformer-based models (fine-tuned BERT). The performance of the models is evaluated using metrics such as accuracy, F1 score, precision, recall, confusion matrix, and ROC-AUC graphs.

Within the scope of this project:

- ComplexANN models were trained using **both randomly initialized embeddings** and **trmodel fine-tune embeddings**. The use of **trmodel fine-tune embeddings** significantly enhanced the performance of ComplexANN, where the accuracy increased from **72.14% (random embeddings)** to **72.24% (trmodel fine-tune embeddings)** due to leveraging semantic context from pre-trained models. **ComplexANN** used an embedding matrix of **400 dimensions**.
- **Transformer-based models** were evaluated using both language-specific models (e.g., savasv/bert-turkish-text-classification) and multilingual BERT models fine-tuned for specific tasks, achieving up to **95.81% accuracy.**

Project Stages



1. ### Data Preprocessing:
   - **Tokenization:** Splitting the text into smaller units (e.g., words or sentences).
   - **Lemmatization:** Converting words to their root forms.
   - **Building Vocabulary:** Creating a vocabulary from the dataset.
   - **Converting to Numerical Representations:** Transforming text into numeric formats suitable for model input.
   - **Padding:** Ensuring all inputs are of the same length, where the input size was standardized to 100 tokens per text.
2. ### Feature Extraction Techniques:
   - **Bag of Words (BoW)**: A basic feature extraction method based on word frequency.
   - **TF-IDF:** A more advanced method that considers word importance in a document relative to the entire corpus.

○ Word2Vec: Converts words into **500-dimensional** vectors to capture semantic information.

3. **Model Training:**

   ○ **Traditional Machine Learning Models**: LightGBM, XGBM, Naive Bayes, SVM-RBF.

   ○ **Neural Networks:** SimpleANN, ComplexANN.

   ○ **ComplexANN** was trained using both **randomly initialized embeddings** and **trmodel fine-tune embeddings,** where **trmodel fine-tune embeddings** utilized a pre-trained embedding matrix of **400 dimensions.**

   ○ **Transformer Models:**

      ▪ **savasv/bert-turkish-text-classification:** A Turkish-specific fine-tuned BERT, achieving an accuracy of **95.81%.**

      ▪ **Bert-tr:** Fine-tuned BERT for Turkish text classification with **92.55%** accuracy.

      ▪ **Bert-multilingual:** A multilingual BERT fine-tuned for general tasks, achieving **91.73%** accuracy**.**

4. **Performance Evaluation:**

   ○ Metrics such as **accuracy, F1 score, precision,** and **recall** were used to assess model performance.

   ○ Confusion matrices and ROC-AUC graphs were analyzed to evaluate classification success. ***The highest F1 score (95.81%)*** was achieved by savasv/bert-turkish-text-classification, while ***the lowest (48.91%)*** *was observed in* ***Naive Bayes with Word2Vec***.

Performance Analysis

Model	Vectorization Method	Accuracy	Precision	Recall	F1 Score
LightGBM	Bag Of Words	78.1633	78.0278	78.1633	77.9841
Xgbm	Bag Of Words	83.3673	83.4370	83.3673	83.3634
Naive Bayes	Bag Of Words	71.2245	72.8540	71.2245	70.6044
SVM-RBF	Bag Of Words	80.8163	80.8804	80.8163	80.7711
SimpleANN	Bag Of Words	83.5714	83.8245	83.5714	83.6045
LightGBM	TF_IDF	77.7551	77.6788	77.7551	77.5897
Xgbm	TF_IDF	82.5510	82.7294	82.5510	82.5669
Naive Bayes	TF_IDF	77.3469	77.6264	77.3469	76.9043
SVM-RBF	TF_IDF	85.6122	85.7023	85.6122	85.6408
SimpleANN	TF_IDF	82.9592	83.2747	82.9592	83.0563
LightGBM	Word2Vec	67.4490	67.7475	67.4490	67.4572
Xgbm	Word2Vec	71.1224	71.1189	71.1224	71.0626
Naive Bayes	Word2Vec	49.5918	51.4999	49.5918	48.9125
SVM-RBF	Word2Vec	58.6735	58.9541	58.6735	58.2359
SimpleANN	Word2Vec	60.5102	62.1591	60.5102	60.3533
ComplexANN	padding	72.1429	73.3368	72.1429	72.0124
ComplexANN	Finetune trmodel	72.2449	73.9738	72.2449	72.4980
Bert-savasy	text-class	95.8163	95.8747	95.8163	95.8116
Bert-tr	finetune	92.5510	92.5940	92.5510	92.5348
Bert-multilingual	finetune	91.7347	91.7939	91.7347	91.6980

1. Bag of Words-Based Models

- **Strengths:**
  - Bag of Words is a simple and effective feature extraction technique for low-complexity tasks.
  - **XGBM** and **SimpleANN** performed well with accuracies of **83.37%** and **83.57%**, respectively, due to their ability to handle large feature spaces and complex patterns.
- **Weaknesses:**
  - **Naive Bayes** achieved only **71.22%** accuracy, struggling due to its independence assumption, which cannot capture complex word relationships.
  - Bag of Words cannot capture word context, limiting its effectiveness in complex datasets.

2. TF-IDF-Based Models

- **Strengths:**
  - TF-IDF captures word importance, resulting in more informative features.
  - **SVM-RBF** achieved the highest accuracy among **TF-IDF** models **(85.61%)** by effectively modeling nonlinear decision boundaries.
- **Weaknesses:**
  - **Naive Bayes,** although improved compared to BoW, still performed poorly with an accuracy of **77.34%,** as it cannot fully utilize the contextual richness of TF-IDF features**.**

3. Word2Vec-Based Models

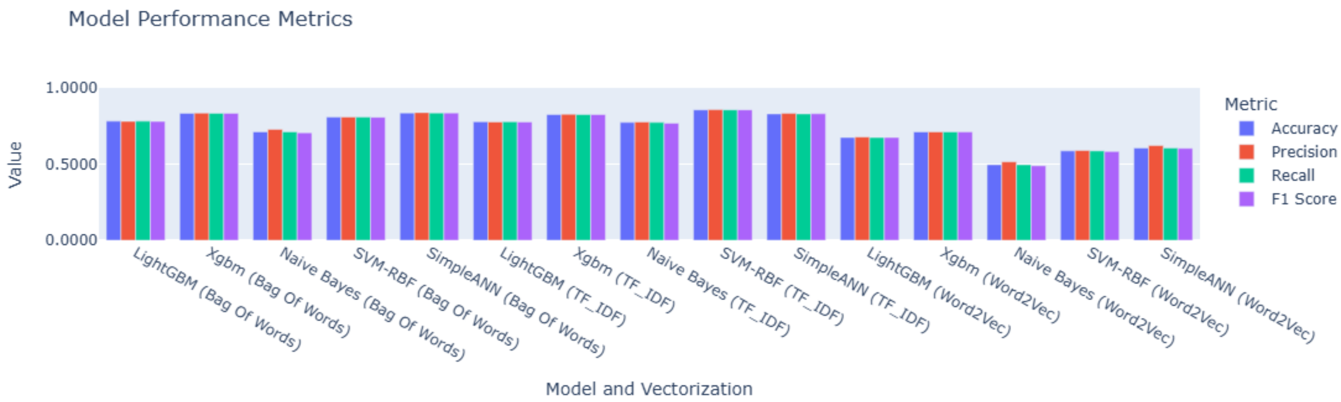
- **Strengths:**
  - **Word2Vec** provides **500-dimensional** semantic embeddings for words, which are beneficial for capturing context.
  - **XGBM** showed relatively better performance with Word2Vec, achieving **71.12%** accuracy due to its ability to handle complex feature interactions.
- **Weaknesses:**
  - **Naive Bayes** and **SVM-RBF** struggled, with accuracies of only **49.59%** and **58.67%,** respectively, as they failed to fully exploit the contextual richness provided by Word2Vec embeddings.

4. ComplexANN Models

- **Strengths:**
  - Leveraging **CNN + BiLSTM, ComplexANN** captures both local and global text patterns.
  - **trmodel fine-tune** embeddings boosted the performance of ComplexANN to an accuracy of **72.24%,** compared **to 72.14%** with random embeddings.
- **Weaknesses:**
  - Limited dataset size **(4,900 samples**) and smaller embedding matrices **(400 dimensions)** restricted the performance of ComplexANN. With larger embedding matrices **(e.g., 768 dimensions)** and more training data, the model could have achieved significantly better results.

5. Transformer Models

- **Strengths:**
  - Transformers excel in capturing complex text relationships using self-attention mechanisms.
  - savasv/bert-turkish-text-classification achieved the highest accuracy **(95.81%)** due to its language-specific optimization for Turkish text.
- **Weaknesses:**
  - Multilingual models like **Bert-multilingual** performed slightly worse **(91.73% accuracy)** due to the lack of specialized fine-tuning for Turkish.



Conclusion

- Transformer-based models, especially **savasv/bert-turkish-text-classification,** achieved the best performance, with an accuracy of **95.81%,** showcasing their superiority in understanding context and complex text relationships.
- Among non-Transformer models, **SVM-RBF** with T**F-IDF** was the most successful, achieving **85.61% accuracy**, due to its ability to handle rich feature spaces and model nonlinear relationships effectively.
- **ComplexANN**, especially when using **trmodel fine-tune embeddings**, showed potential with an accuracy **of 72.24%.** However, its performance was limited by the dataset size **(4,900 samples**) and embedding matrix dimensions **(400 dimensions).** With larger embedding matrices **(e.g., 768 dimensions)** and more training data, ComplexANN could rival Transformer models in performance.