Name: Mohamed Nawas Raza Mohamed

Student Number: 24148501

Link to Code: https://colab.research.google.com/drive/1c-eMmWZyJmEavsN43SDiKr__Py0QRGx5?usp=sharing

Offensive Content Detection: A Critical Analysis

In this project, I developed an NLP system to detect offensive content on social media. I began by exploring the dataset through visualizations—plotting tweet length distributions and label frequencies using histograms and word clouds. These insights revealed significant variability and class imbalance, which necessitated rigorous data preprocessing.

Data Processing and Feature Extraction

Text was first normalized by converting it to lowercase and then cleaned by removing URLs, user mentions, hashtags, punctuation, and numbers. I further refined the data by eliminating stopwords and applying lemmatization using spaCy. The cleaned tweets were then transformed into numerical features using TF-IDF vectorization with up to 5000 features and bi-grams:

```
vectorizer = TfidfVectorizer(max_features=5000, ngram_range=(1, 2))
X = vectorizer.fit_transform(train_df['clean_tweet'])
X_test = vectorizer.transform(test_df['clean_tweet'])
```

Modeling and Ensemble Techniques

I experimented with a variety of models, including:

- **CatBoost (CB)**
- **ExtraTrees (ET)**
- **RandomForest (RF)**
- **XGBoost (XGB)**
- **Logistic Regression (LR)**
- **LightGBM (LGBM)**
- **SVC**
- **Naïve Bayes (NB)**
- **Gradient Boosting (GB)**
- **AdaBoost (AB)**
- **MLP**

To leverage their strengths and mitigate individual weaknesses, I built ensemble models using the VotingClassifier. The ensembles combined predictions from ExtraTrees, RandomForest, XGBoost, and a custom CatBoost wrapper. The **Voting Soft** ensemble, which averages probabilistic predictions, achieved the best balance between training and validation performance:

```
voting_soft = VotingClassifier(
    estimators=[
        ('et', ExtraTreesClassifier(n_estimators=100, random_state=42)),
        ('rf', RandomForestClassifier(n_estimators=100, random_state=42)),
        ('xgb', XGBClassifier(eval_metric='mlogloss', random_state=42)),
        ('cb', CatBoostWrapper(random_state=42))
    ],
    voting='soft'
)
```

Model Evaluation & Comparison

We tested Random Forest, CatBoost, XGBoost, and Voting ensembles (Hard, Soft). Below is our results table, which shows Voting Soft achieving the highest validation accuracy due to blending diverse predictive strengths.

	Train Accuracy	Val Accuracy	Acc Diff	Precision	Recall	F1-Score
CB	0.797907	0.747519	0.050387	0.712428	0.747519	0.705831
ET	0.991449	0.741693	0.249756	0.713902	0.741693	0.718713
RF	0.991449	0.750217	0.241232	0.724878	0.750217	0.716115
XGB	0.845382	0.74547	0.099912	0.720765	0.74547	0.708483
LR	0.817329	0.73716	0.080168	0.698854	0.73716	0.694426
LGBM	0.847054	0.730472	0.116582	0.697117	0.730472	0.69754
SVC	0.911955	0.725723	0.186232	0.695813	0.725723	0.702873
NB	0.759171	0.715581	0.043591	0.6842	0.715581	0.642434
GB	0.770285	0.738671	0.031614	0.718552	0.738671	0.68628
AB	0.725966	0.720544	0.005422	0.691888	0.720544	0.657392
MLP	0.991395	0.66735	0.324045	0.660512	0.66735	0.663401
Voting Hard	0.857116	0.749677	0.107438	0.731991	0.749677	0.706811
Voting Soft	0.984436	0.755289	0.229147	0.734409	0.755289	0.720864

Best Model Selection

Based on extensive 5-fold cross-validation, the Voting Soft ensemble emerged as the best approach due to its balanced performance and generalizability. The final notebook includes the code for this model only, ensuring a focused and reproducible implementation.

Conclusion

By cleaning tweets, applying TF-IDF, and using a Voting Soft ensemble, we achieve high accuracy in detecting offensive content. While AI can't fully replace human oversight, it significantly reduces toxicity and fosters safer online interactions. For future work, deep learning architectures (e.g., BERT) or sentiment analysis could enhance detection capabilities and context understanding even further.