**Instagram Influencer Category Prediction**

**Overview**

This repository contains the code and resources for a project aimed at classifying Instagram influencers into predefined categories based on the content of their posts and profile information. The project leverages various machine learning and natural language processing (NLP) techniques.

The repository includes:

**classification.py:** This Python script contains the core logic of the project, including:

Data loading and preprocessing functions for both training and test datasets.

Implementation of a custom Turkish text preprocessor, `TurkishTextPreprocessor`, that handles:

* Irrelevant content removal (HTML, URLs, excessive whitespace).
* Normalization of Turkish characters.
* Lowercase conversion.
* Tokenization using Zemberek.
* Stop word removal.
* Lemmatization.
* Emoji to text conversion.
* Hashtag and mention handling.

Data loading for the training dataset from /content/drive/My Drive/Project/training-dataset.jsonl.gz` and the labels from `/content/drive/My Drive/Project/train-classification.csv`.

Implementation of TF-IDF vectorization and model training using several machine learning algorithms:

* Support Vector Classifier (SVC) with hyperparameter tuning.
* LightGBM classifier.
* Logistic Regression with hyperparameter tuning.
* Random Forest classifier with hyperparameter tuning.
* Sentence Transformer embeddings with K-means clustering and a Random Forest classifier.
* Multi-Layer Perceptron (MLP) classifier with SMOTE and without.
* Evaluation metrics (accuracy, classification report, and confusion matrix).
* Integration with Google Drive for data access and model persistence.

**Methodology**

Our approach involves several key steps:

**Data Acquisition**

The training dataset is loaded from a gzipped JSONL file, which contains user profile information and their posts. The label information is loaded from a separate CSV file. Data is split into training and test sets based on the availability of categories and labels in the training data, where if a username exists in the category labels it is considered train data, if not its considered as test data.

**Preprocessing**

Preprocessing is critical for ensuring the quality and consistency of input data. The following steps were taken:

* **Text Normalization:**
* Standardized Turkish-specific characters such as "ý" to "ı" and "þ" to "ş" to maintain linguistic accuracy.
* **Tokenization:**
* Applied Zemberek's tokenizer to split text into meaningful tokens, handling punctuation and special symbols appropriately.
* **Stop-Word Removal:**
* Utilized a curated list of Turkish stop-words, customized to exclude common but non-informative terms in the dataset.
* **Emoji and Mention Handling:**
* Converted emojis to descriptive text to retain emotional and contextual nuances.
* Managed mentions by either retaining, removing, or treating them as regular tokens, depending on the task requirements.
* **Lemmatization:**
* Leveraged Zemberek's morphological analyzer to extract base forms of words, improving generalization and reducing sparsity.

**Feature Extraction**

* **TF-IDF Vectorization:**
* Used term frequency-inverse document frequency (TF-IDF) to represent text data as weighted numerical vectors.
* Configured n-grams (up to bi-grams) to capture contextual relationships between words.
* **Embedding Generation:**
* Applied SentenceTransformers to create dense, high-dimensional embeddings for textual data, capturing semantic meaning effectively.

**Classification**

* **Model Selection and Training:**
* Experimented with multiple classifiers (SVC, Random Forest, LightGBM, and Logistic Regression) to identify the best-performing model.
* Balanced class weights to address the issue of class imbalance in the dataset.
* **Deep Learning Approach:**
* Fine-tuned a multilingual RoBERTa model for sequence classification tasks, adapting it for the Turkish language.
* **Hyperparameter Optimization:**
* Employed GridSearchCV to fine-tune model parameters, improving accuracy and generalization.

**Clustering**

* **K-Means Clustering:**
* Grouped similar text data into clusters using K-Means, with the number of clusters informed by the unique categories in the dataset.
* **Cluster Evaluation:**
* Analyzed cluster purity by comparing majority categories within each cluster to true labels.

**Evaluation**

* **Performance Metrics:**
* Evaluated models using accuracy, precision, recall, F1-score, and confusion matrices.
* **Visualization:**
* Plotted feature importance and confusion matrices to interpret model behavior and identify areas of improvement.

**Results**

**Experimental Findings**

* **Accuracy Metrics:**
* SVC: Achieved validation accuracy of 67% with balanced class weights.
* LightGBM: Recorded a validation accuracy of 63% with optimal hyperparameters with balanced class weights.
* RoBERTa: We have used FacebookAI/xlm-roberta-base and its accuracy is 70%. Also we have used burakaytan/roberta-base-turkish-uncased and it reached accuracy of 67%. Cross entropy loss and balanced class weights are used.
* RandomForest: Reached a validation accuracy of 64% with balanced class weights.
* Logistic Regression: Achieved validation accuracy of 68% with hyperparameter tuning with balanced class weights.
* CatBoost: Recorded a validation accuracy of 62%.
* MLP: Achieved validation accuracy of 65% with MLP, when we applied SMOTE it did not improve and decreased to accuracy of 62%.

**Results of SVC classification:** SVC reached best cross validation accuracy among all models with 67% accuracy rate and used for ROUND3 classification.



**Results of RoBERTa classification:** Eventhough it is not as robust as SVC, RandomForest, or Logistic Regression, it has given 70% accuracy on validation. Therefore, we have used for classification ROUND1(FacebookAI/xlm-roberta-base) and ROUND2(burakaytan/roberta-base-turkish-uncased).

**Instagram Like Count Prediction**

**Overview:**

For the regression task of the project, multiple models were built in every round to achieve greater performance metrics. The features which will be explained shortly were the same throughout the project. The preprocessing pipeline was designed to handle data cleaning, feature engineering, and user-level aggregation effectively. Key transformations were implemented, including logarithmic scaling of follower and comment counts, alongside categorical encoding of business account status. Particular attention was given to missing value handling, ensuring data integrity throughout the process. The distribution analysis identified highly skewed patterns in key metrics. The findings revealed strong correlations between average like counts and follower patterns, while comment count distributions provided valuable insights into user engagement behavior. These distributional challenges were addressed through strategic feature transformations. In each of every 3 rounds, different methods tried to improve the model’s performance. These methods will be explained in the following.

**Round 1:**

In the first round, a dual-model system was developed to handle both regular and high-engagement accounts. The system demonstrated strong predictive performance, particularly for high-engagement outlier accounts. The Random Forest model, configured with 150 estimators and a max depth of 10, achieved a Log MSE of 0.3640 on training data and 0.8326 on validation data for regular accounts. For high-engagement outliers, a Ridge Regression model was implemented, which demonstrated superior performance with a Log MSE of 0.3181 on training data and 0.4025 on validation data, indicating stronger generalization capabilities. Account segmentation was established using 95th percentile thresholds, with separate processing pipelines implemented for each segment using an 85%-15% train-validation split. The prediction pipeline incorporated logarithmic transformations of target variables and maintained separate processing streams for regular and outlier accounts. To ensure prediction quality, maximum value constraints were implemented to prevent negative predictions, with inverse log transformations applied for final outputs. The output is generated in JSON format with user IDs and predicted like counts.

**Round 2:**

In the second round, unlike the first round, a clustering-based system was developed to handle different user engagement patterns. The system demonstrated enhanced predictive performance through the implementation of three distinct clusters. The K-means clustering algorithm segmented the data into three groups, containing 2,166, 1,009, and 2,216 users respectively. For each cluster, specialized Random Forest Regressors were implemented with varying configurations. The first and third clusters utilized 150 estimators with a max depth of 6, while the second cluster employed 100 estimators with the same depth constraint. The models showed varying performance across clusters, with Cluster 0 achieving a mean MSE of 0.696 and test log error of 219.39, Cluster 1 demonstrating a mean MSE of 0.863 and test log error of 100.36, and Cluster 2 exhibiting the strongest performance with a mean MSE of 0.549 and test log error of 189.37. The prediction pipeline incorporated cluster-specific processing, with test instances assigned to appropriate clusters before prediction. The system maintained data quality through logarithmic transformations of key metrics including follower count, average like count, and comment count sum. Post-processing steps ensured prediction validity through non-negative value enforcement and inverse log transformation. The final predictions were generated in JSON format, maintaining consistency with the first round's output structure while leveraging the enhanced accuracy provided by the cluster-specific approach.

**Round 3:**

In the third round, an enhanced clustering-based system was developed using a combination of DBSCAN and K-means algorithms to improve prediction accuracy. The initial dataset of 5,391 users underwent preliminary outlier removal using DBSCAN, resulting in a refined dataset of 5,342 users, demonstrating the system's robust approach to handling extreme cases. The K-means algorithm then segmented these users into three distinct clusters containing 2,264, 2,094, and 984 users respectively, providing a more nuanced categorization of user engagement patterns.

The model architecture implemented specialized Random Forest Regressors for each cluster, all configured with 150 estimators and a max depth of 6. The models demonstrated exceptional performance across clusters, with Cluster 0 achieving a mean MSE of 0.00165 in 10-fold cross-validation and a test log error of 1.596, Cluster 1 showing a mean MSE of 0.00169 and test log error of 2.121, and Cluster 2 recording a mean MSE of 0.00252 and test log error of 2.656. These metrics represent a significant improvement over previous rounds, particularly in the handling of different user engagement patterns.

The prediction pipeline maintained sophisticated data processing, with logarithmic transformations applied to key metrics including follower count, average like count, and comment count sum. Test data processing involved assigning instances to appropriate clusters, with 1,237 users assigned to Cluster 0, 1,176 to Cluster 1, and 552 to Cluster 2. The system ensured prediction quality through non-negative value enforcement and inverse log transformation, with final predictions generated in a structured JSON format.

This third iteration demonstrated substantial improvements in prediction accuracy through its sophisticated outlier handling and cluster-specific modeling approach. The combination of DBSCAN for initial outlier removal and K-means for user segmentation provided a more refined understanding of user engagement patterns, leading to more accurate and nuanced predictions across different user categories.

**Comperative Analysis:**

The evolution across rounds shows a clear progression in prediction accuracy:

Round 1 established the importance of separate handling for different engagement levels, with MSE values in the 0.3-0.8 range.

Round 2 refined this approach through clustering, though with higher MSE values, it provided more nuanced predictions across user segments.

Round 3 achieved the most precise predictions with MSE values in the 0.001-0.002 range, representing an order of magnitude improvement over previous rounds.

The progression demonstrates the effectiveness of increasingly sophisticated clustering approaches, with the final round's combination of DBSCAN and K-means providing the most accurate predictions. The significant reduction in error metrics from Round 1 to Round 3 validates the evolution of the methodology, particularly in handling different user engagement patterns and outlier management.

These results suggest that while the initial dual-model approach provided a solid foundation, the refined clustering methods of later rounds were crucial in achieving superior prediction accuracy. The final implementation's sophisticated outlier handling and cluster-specific modeling approach represents the most effective solution for Instagram engagement prediction among the three rounds.

**Team Contributions**

* **[Hüseyin Doğan Türk (Classification)]:**
* Designed and implemented the text preprocessing pipeline.
* Curated the Turkish stop-word list and emoji mappings.
* Conducted deep learning experiments using RoBERTa.
* Integrated SentenceTransformers and implemented clustering algorithms.
* Evaluated cluster purity and refined embedding techniques.
* **[Süleyman Berber (Classification)]:**
* Developed scripts for TF-IDF vectorization and training classification models.
* Designed and implemented the text preprocessing pipeline.
* Conducted hyperparameter tuning and model optimization.
* Conducted deep learning experiments using RoBERTa.
* Managed evaluation metrics and generated visualizations for results.
* **[İlhan Sertelli (Regression)]:**
* Developed the dataset preprocessing.
* Applied visualizations and clustering.
* Helped in training the dataset.
* Applied cross-validation operations in order to improve the performance.
* Helped with the feature selection.
* **[Atacan Dilber (Regression)]:**
* Executed feature selection and correlation analysis
* Helped to implement Random Forest and Ridge Regression models
* Generated and validated prediction outputs
* Helped in applying K-means and DBSCAN clustering
* Developed preprocessing pipeline and implemented feature engineering